

MidiPlay

Janne Syväniemi

COLLABORATORS

	<i>TITLE :</i> MidiPlay		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Janne Syväniemi	January 19, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MidiPlay	1
1.1	MidiPlay	1
1.2	INFORMATION	1
1.3	INTRODUCTION	2
1.4	NEW FEATURES	5
1.5	INSTALLATION	7
1.6	STARTING MIDIPLAY	8
1.7	STARTING FROM CLI	8
1.8	STARTING FROM WORKBENCH	11
1.9	HELP	14
1.10	MIDI FILE INFORMATION	14
1.11	META EVENT PRINTING	14
1.12	DO NOT SEND MIDI EVENTS	15
1.13	SET TEMPO	15
1.14	MIDI EVENT PRINTING	15
1.15	MUTE CHANNELS	16
1.16	SOLO CHANNEL	16
1.17	TRANSPOSE	16
1.18	SETS DRUM CHANNEL	17
1.19	PROGRAM CHANGE FILTER	17
1.20	PITCH BENDER FILTER	17
1.21	LOOP	18
1.22	SETS PLAYER PRIORITY	18
1.23	SEQUENTIAL PLAYING	18
1.24	MIDI MERGE	19
1.25	EXTERNAL SYNC	19
1.26	CIA TIMER	19
1.27	SHOWS MAIN WINDOW	20
1.28	SKIPS EMPTY MEASURES	20
1.29	NO PROGRAM CHANGE ON DRUM CHANNEL	20

1.30 NO MAPPING ON CHANNELS	21
1.31 BUFFER AS DEFAULT SELECTION	21
1.32 IMMEDIATE START AFTER LOADING	22
1.33 EXIT AFTER PLAYING	22
1.34 WAITS EXTERNAL START	22
1.35 MAPPING	23
1.36 PROGRAM CHANGE MAPFILE	23
1.37 NORMAL RESET	23
1.38 QUANTIZE	24
1.39 PLAYLIST FILE	24
1.40 READS TOOLTYPES	24
1.41 MELODY CHANNEL	25
1.42 DRUM MAPFILE	26
1.43 SOFT MUTING	26
1.44 COUNT MEASURES AND COLLECTS TEXT EVENTS	27
1.45 MEASURE FIELD STATUS	27
1.46 FADE	28
1.47 LYRIC TYPE	28
1.48 SETS PLAYLIST PLAYING	28
1.49 CHANNEL PATCHING	29
1.50 VOLUMECHANGE	29
1.51 DEFINES SCRIPT FOR MESSAGES	30
1.52 DEFINES SCRIPT FOR FILENAMES	30
1.53 DEFINES SCRIPT FOR NON-MIDI FILES	31
1.54 FORCES RESET	31
1.55 REMEMBERS LAST DIRECTORY	31
1.56 CHANNELBYTE FOR TONE SYSEX	32
1.57 TYPE OF BANK CHANGE MESSAGE	32
1.58 SYSEX MESSAGES IN FILES NOT SENT	33
1.59 PROGRAM CHANGE FOR CURRENT FILE	33
1.60 CONNECTS AN EXTERNAL MIXER	34
1.61 TOGGLES SELECTED CHANNEL	34
1.62 TOGGLES SELECTED CHANNEL	35
1.63 ACTIVATES THE NEXT OPEN WINDOW	35
1.64 ACTIVATES THE DRUM CHANNEL GADGET	35
1.65 ACTIVATES THE SOLO CHANNEL GADGET	36
1.66 ACTIVATES THE MELODY CHANNEL GADGET	36
1.67 VOLUME CHANGE OF THE MELODY CHANNEL DOWN	36
1.68 VOLUME CHANGE OF THE MELODY CHANNEL UP	36

1.69	LOADS KEYBOARD SETTINGS	37
1.70	RETURNS DEFAULT KEYBOARD	37
1.71	DEFINES PATTERN USED IN MIDI FILE REQUESTER	38
1.72	DEFINES CONFIGURATION FILENAME	38
1.73	DEFINES THE NAME OF MIDIPLAY'S AREXX PORT	38
1.74	ICONIFIES OR DEICONIFIES MIDIPLAY	39
1.75	MIDI START/CLOCK/STOP MESSAGES WILL BE SENT	39
1.76	SENDS NOTEON MESSAGE	39
1.77	SENDS NOTEOFF MESSAGE	40
1.78	PROGRAM CHANGE MAPS FOR DIFFERENT CHANNELS	40
1.79	INTROPLAY	42
1.80	INTROTIME	42
1.81	REQTOOLS REQUESTERS WILL BE USED INSTEAD OF ASL	42
1.82	MIDIPLAY OPENED ON AN EXISTING PUBLIC SCREEN	43
1.83	FONT FOR THE LYRIC WINDOW	43
1.84	COLORS FOR THE LYRICS	43
1.85	SELECT BY NUMBER FROM PLAYLIST	44
1.86	SELECTION TO BUFFER	44
1.87	PLAYLIST LOCKED	45
1.88	ALL NOTES OFF MESSAGE SENT	45
1.89	PLAY COMMAND	45
1.90	STOP COMMAND	45
1.91	CONTINUE PLAYING COMMAND	46
1.92	LOADS FILE	46
1.93	LOADS FILE, NOT ADDED TO THE PLAYLIST	46
1.94	ADDS FILE TO THE PLAYLIST	47
1.95	TRANSPOSE UP	47
1.96	TRANSPOSE DOWN	47
1.97	TEMPO UP	48
1.98	TEMPO DOWN	48
1.99	QUITS MIDIPLAY	48
1.100	NEW PLAYLIST	48
1.101	LOADS PLAYLIST	49
1.102	FILTER FOR TEMPO CHANGES	49
1.103	OPENS/CLOSE THE LYRIC WINDOW	49
1.104	OPENS (CLOSES) THE SELECTION WINDOW	50
1.105	MUTES/UNMUTES THE MELODY CHANNEL	50
1.106	SELECTS NEXT NUMBER FROM BUFFER	50
1.107	CLEAR THE BUFFER	51

1.108	FAST FORWARD	51
1.109	VOLUME UP	51
1.110	VOLUME DOWN	52
1.111	SYSTEM EXCLUSIVE COMMAND	52
1.112	SEND NORMAL MIDI DATA	52
1.113	MIDI MERGE TRANPOSE VALUE	53
1.114	RANDOM PLAY	53
1.115	PLAY THE FIRST FILE IN THE PLAYLIST	54
1.116	PLAY THE NEXT FILE IN THE PLAYLIST	54
1.117	PLAY THE PREVIOUS FILE IN THE PLAYLIST	54
1.118	RELATIVE TRANSPOSE WITH FILES	55
1.119	DEFAULTS FOR FILES	55
1.120	DEFAULT DIRECTORY FOR MIDI FILE	55
1.121	DIRECTORY PATH FOR THE MIXER	56
1.122	USER DEFINABLE APPLICATION ICON	56
1.123	NAME SHOWN UNDER APPICON	56
1.124	APPICON POSITION	57
1.125	QUERY FOR SETTINGS OF MIDIPLAY	57
1.126	Example status script	61
1.127	WINDOWS	61
1.128	MAIN WINDOW	62
1.129	MENUS	62
1.130	PROJECT	62
1.131	FUNCTIONS	63
1.132	SETUP MENU	64
1.133	MIDI OPTIONS	64
1.134	RESET OPTIONS	65
1.135	PLAY OPTIONS	65
1.136	FILTER OPTIONS	65
1.137	SHOW OPTIONS	66
1.138	MISC OPTIONS	66
1.139	MUTE ON	66
1.140	SOLO ON	67
1.141	RESET	67
1.142	BUFFER	67
1.143	SELECTION WINDOW	68
1.144	MAPFILE	69
1.145	EXAMPLE MAP FILE	69
1.146	FILE CONTROL	72

1.147WINDOWS	72
1.148GADGETS	72
1.149KEYBOARD	74
1.150PREFERENCES WINDOW	75
1.151MELODY CHANNEL	75
1.152PATCH CHANNELS WINDOW	76
1.153TRACK NAME WINDOW	76
1.154PLAYLIST WINDOW	76
1.155LYRIC WINDOW	77
1.156AREXX COMMANDS	78
1.157OTHER INFORMATION	83
1.158TODO	83
1.159PROBLEMS AND LIMITATIONS	83
1.160HISTORY	84
1.161AUTHOR	93
1.162CREDITS	94
1.163TIPS FOR USERS	94
1.164EXAMPLE 1	95
1.165EXAMPLE 2	95
1.166EXAMPLE 3	96
1.167MIDIPLAY TOOLKIT	97
1.168How to adjust MIDI files using icons	98
1.169Melody channel	99
1.170Transposing	100
1.171Tempo changing	100
1.172Program change on a channel	101
1.173Channel patching	102
1.174Volume Changing	103

Chapter 1

MidiPlay

1.1 MidiPlay

INFORMATION

INTRODUCTION

INSTALLATION

NEW FEATURES

STARTING OPTIONS

WINDOWS

AREXX COMMANDS

TIPS FOR USERS

TUTORIAL FOR ADJUSTING MIDI FILES USING ICONS

OTHER INFORMATION

TODO

PROBLEMS AND LIMITATIONS

HISTORY

AUTHOR

CREDITS

MIDIPLAY TOOLKIT

1.2 INFORMATION

VERSION:

MidiPlay v2.63b 20-Nov-95

If you need the latest version of MidiPlay, look at my WWW Amiga page:
<http://www.uta.fi/~csjasy/amiga.html>

Unfortunately there is currently no support BBS.

REQUIREMENTS

MidiPlay requires at least OS 2.04 to run. Also midi.library and miditune.library are required. If you want to really hear something, you need external MIDI equipments and a MIDI file.

DISCLAIMER

The author of this program can not be held responsible for any damage directly or indirectly caused by the use of this program. Use it at your own risk.

COPYRIGHT

This program is copyright (C) 1993-1995 Janne Syväniemi. All Rights Reserved. The program is GIFTWARE. It is freely distributable as long as all files are included in their original form and no extra fee is charged. MidiPlay may not be included in any commercial package without a permission of the author. People who use MidiPlay much, should consider rewarding the author with some music related gift (like good music in CD format). If you support MidiPlay, you will receive
MidiPlay ToolKit
.

miditune.library is copyright (C) 1995 Janne Syväniemi. All Rights Reserved. Distribution separately from the MidiPlay archive is not allowed without a written permission of the author.

Bill Barton's midi.library is Copyright (C) 1987, 1988, Pregnant Badger Music. More info about midi.library can be found on Fred Fish disk 227.

reqtools.library is copyright (C) Nico François.

See also

Author

1.3 INTRODUCTION

WHAT IS MIDIPLAY?

MidiPlay is a program that can play format 0 and 1 MIDI files. It can also

print their contents.

You can change most of the features of MidiPlay via a graphical user interface. You can mute instruments or let them play solo. You can play files in different key by transposing. You can speed up or slow down playing by changing the tempo. You can filter certain MIDI messages e.g. pitch bender, program change or system exclusive messages. You can synchronize playing from an external source. You can merge your keyboard playing with MIDI merging.

An ARexx port provides even better control for fine tuning of MidiPlay's performance. MidiPlay has over 100 different ARexx commands.

You can also change the playing of individual MIDI files via icon tool types.

You can add songs to a playlist. You can select songs easily from the playlist with a single click. A file buffer makes it easy to select many songs from the playlist at once to be played next.

MIDI files and General MIDI files

MIDI files are usually made on one particular system. These files won't necessarily sound good on your system, you may have a completely different synth which has instruments ordered differently than expected in a MIDI file. General MIDI standard tries to remove this problem, but many old files are still made before this standard and many people don't have GM synths. Changing all your MIDI files' instrumentation to correspond your setup by using a sequencer program would require a lot of work and if you changed your setup, you would have to do the whole work again. MidiPlay tries to prevent these problems with many program change mapping features.

If you have a GM synth and you listen to GM files you won't have many problems if any. But if you have a synth which doesn't support GM standard, you should make a mapfile for your synth. MidiPlay support two kind of map files. One for normal program change mapping and the other for drum mapping (i.e. for individual note on events on the drum channel). In fact, you can have different map files for every MIDI channel. For example, GM standard says that a program change number 25 corresponds to a nylon guitar, but the nylon guitar can be set with PC number 69 on your synth. In this case, you should put number 69 to your map file's 25th row. If your synth supports different banks for an individual instrument, you can exploit this feature in your map file. It is also possible to send a complete instrument setup via a sysex message file. See the exact

description of the map file

.

The following commands also affect the instrument mapping:

```
set mapping on/off: MAP
tell the instrument map file:
    MAPFILE
    tell the drum map file: DRUMMAPFILE
tell the mapping on an individual channel:
    CHANNELMAP
```

```
filter all program change messages:
  NOPC
tell the channels not requiring mapping:
  NOMAPCHANNELS
set an instrument for a channel:
  CHANNELPC
```

Even if you have a GM synth, you may want to change the playing of the files beforehand without editing the files with a sequencer program. You can use tool types in a file's icon for this purpose. For example, if there is a very long silence after the last audible event before the end of the song, you can use

```
SKIP
tool type for defining the longest
```

gap between events and if a longer gap is encountered it will be skipped immediately. You can also use commands like TRANSPOSE to play your song in different key or command NOPB to filter out the pitch bender messages. If you are not satisfied with these settings, you only have to edit these tool types.

Because these changes are not necessarily changed back when the next file is loaded, you can use a default file defined with FILEDEFAULTS command to set all your default file values, which are set before the next file's tool types are read. With this feature you can have default values for files which have no tooltypes or e.g. tempo or instruments defined. You might have also changed (e.g. muted channels) your settings via the user interface during the playing of the previous song, but when this default file is loaded e.g. with MUTE=0 as a tooltype, all channels will be unmuted and you don't have to make all these changes by hand. See

```
ICON
option for more information.
```

See also

```
TUTORIAL FOR ADJUSTING MIDI FILES USING ICONS
provided
```

by Johnny Bonde.

Karaoke MIDI files

MidiPlay supports also files which have lyrics. MidiPlay can show certain amount of rows in a lyric window. The word to be sung is highlighted. The most common lyric format is soft karaoke which includes certain formatting commands, e.g. new line and new paragraph. These commands are known by MidiPlay and MidiPlay tries to show the lyrics according to them. There are also MIDI files which have no special format but all the lyrics are in these files as lyric events i.e. like the MIDI file standard says. MidiPlay can handle this case too by showing as much lyric events in the lyric window as possible (highlighting will work also).

You can change the look of the lyric window with the following commands:

```
LYRICFONT
```

```
LYRICCOLORS
```

Other features include:

- o with MESSAGEREXX, FILENAMEREXX and FILETYPEREXX parameters you can e.g. hear messages and filenames via say command, view text files associated with MIDI files or decompress archives and play MIDI files in them

See also

NEW FEATURES

.

1.4 NEW FEATURES

NEW FEATURES IN 2.63

- o new
 - Patch Channel window
- o new play button images in the main window
- o
 - FILEPATH command for defining the default MIDI file directory

NEW FEATURES IN 2.61

- o def_midisplay.info replaced with user definable FILEDEFAULTS file.
 - o you can now use <file>.cfg instead of tooltypes in file's icon. See option
 - ICON for more information.
 - o
 - RELTRANSPOSE option for compatibility with older versions of MidiPlay
 - o playlist can now have its own configuration file (or tooltypes in its icon)
 - o most of the UP/DOWN commands available now also in configuration files
 - o new commands:
 - MIXERPATH
 - '
 - APPICONFILE
 - '
 - APPICONNAME
 - '
 - APPICONPOS
- NEW FEATURES IN 2.59

- o STATUS
command for asking most of the settings of MidiPlay
 - NEW FEATURES IN 2.58:
 - o all loading and playing routines now in a shared library called miditune.library
 - o reqtools.library now optional, ASL file requester used as default.
See option REQTOOLS
 - o better support for karaoke files, lyric window changed: ↔
resizable,
no more lyrictype button, more rows of lyric shown, also those words shown that should be sung later and the current word highlighted. The best support the soft karaoke files, but works also with files which have normal lyric events. You can control the font and the colors of the lyric window with options LYRICFONT
and
LYRICCOLORS
.
 - o CHANNELMAP
features lets you make program change maps also for individual channels
 - o new kind of selection window
with name selection also
 - o menus
reorganized
 - o Playlist window
changed: resizable, no more path text gadget, edit gadget opens now a file requester, Select => Play now called Lock
 - o Text Event Window now
Track Name Window
, shows only
tracknames of the current MIDI file, no more check box gadgets
 - o Intro play feature which lets you play only few seconds of each file
See INTROPLAY
and
INTROPLAYTIME
 - o MidiPlay can be opened on a public screen with PUBSCREEN
option
 - o You can transpose all note on/off messages to be merged by a certain amount
-

with
 MERGETRANSPOSE
 command

o With
 SENDMIDIDATA
 you can transmit different MIDI
 messages to MIDI out.

o Other new commands:
 SELECTNUMBER
 ,
 SELECTBUFFER
 NEW FEATURES IN 2.30:

o option
 MIDICLOCK
 (AREXX command SETMIDICLOCK)

NEW FEATURES IN 2.26:

o Configuration file will be loaded as ENV:MidiPlay/MidiPlay.prefs
 or MidiPlay.prefs unless defined with SETTINGS=filename tooltype!
 You MUST define

 SETTINGS
 =S:midisplay.config, if you want to use your
 old version, or rename it to MidiPlay.prefs

o Window positions with open/closed and zoom information can be saved

o
 NOTEON
 and
 NOTEOFF
 commands

o
 PORTNAME
 tooltype defines the name of the MidiPlay's AREXX port

o FORWARD command requires now a value!

o AppWindow and AppIcon features

1.5 INSTALLATION

Copy libraries midi.library and miditune.library to your
 LIBS: directory. A template configuration file
 (midisplay.prefs) is included with this archive, edit it to
 suit your needs and copy it to your MidiPlay directory, or
 ENV:MidiPlay and ENVARC:MidiPlay directories or define
 with SETTINGS=filename where it is located.

Copy MidiPlay wherever you like. Configuration file is

not necessary, you can use the tooltypes in the MidiPlay icon or in the project icons.

If you use map files you should define in the configuration file where they are located.

Couple of example ARexx scripts are included, copy them to your REXX: directory, if you want to use them. See starting options

```
FILENAMEREXX
```

```
,
```

```
FILETYPEREXX
```

```
and
```

```
MESSAGEREXX
```

```
for more information how to use them.
```

You can use ReqTools requesters from the reqtools.library. This library is not included. See option

```
ReqTools
```

```
.
```

1.6 STARTING MIDIPLAY

Before the starting options or tooltypes are read, MidiPlay looks for configuration file name in SETTINGS=filename or +0filename (CLI argument). If the file name is not defined, MidiPlay looks for MidiPlay.prefs in the current directory or in ENV:MidiPlay directory. The first file found will be loaded. If the configuration file is not found or defined, MidiPlay's default settings are used. All configuration file settings can be changed through CLI arguments or tooltypes.

You can begin a row in the configuration file with a '#' mark to tell to MidiPlay it is a comment row, thus rest of the row will be ignored.

From

```
CLI
```

```
From
```

```
WORKBENCH
```

1.7 STARTING FROM CLI

```
TYPE: midiplay [+|- options] [filename]
```

NOTE (v2.0) : '-' sign means not the same as in previous versions! '+' sign means now that this options is set ON and '-' sign means that this option is set OFF. There is

a '*' sign in the following list if + or - has no effect (you can use both of them).

NOTE (v2.20): Symbol '|' is still available but using symbol ',' is now recommended.

NOTE (v2.21): '-' option is now meaningful only with toggle options. This means that M (melody), m (mute), s (solo) and G (mapfile) have changed. '-' sign can be put right after the option letters m or s to get the same effect as before. Mapping can be set on/off using K option.

NOTE (v2.21 & 2.60): if option ICON is set, MidiPlay tries to read default values from a file default file before loading tooltypes of a MIDI file. See options

```

ICON
and
FILEDEFAULTS
for more information.

```

NOTE (v2.26): SETTINGS=filename defines now the correct configuration file name, S:midisplay.config isn't known to MidiPlay anymore. midisplay.prefs in ENV:MidiPlay directory is recommended.

More than one file is allowed. If no filename has been given, MidiPlay expects input from the current input stream, so redirection can be used.

Simple options can be given together like: +dcbl. Options with numbers or text must be given separate from the following options. For example: +u10 +lbp35 +m1 +m2

```

HELP      :
          *?
          FILEINFO      :
          +f
          TEXTINFO      :
          +x
          NOMIDIOUTPUT:
          +n
          TEMPO          :
          *tVALUE
          DATAINFO      :
          +d
          MUTECHANNELS:
          *mC1,C2...
          SOLOCHANNEL    :
          *sC
          TRANSPOSE      :
          *rVALUE
          DRUMCHANNEL    :
          *uC
          PC FILTER      :

```



```
+c
  PB FILTER      :
+b
  LOOP          :
+l
  PRIORITY      :
*pVALUE
  SEQUENTIAL    :
+q
  MERGE         :
+g
  EXT. SYNC     :
+e
  CIA TIMER     :
+i
  MAIN WINDOW   :
+w
  SKIP          :
*kVALUE
  DRUM PC FILT :
+o
  NOMAPCHANNEL  :
*hC1,C2...
  BUFDEFAULT    :
+a
  START         :
+S
  EXIT         :
+X
  MIDISTART     :
+D
  MAPFILE      :
*Gfile
  MAP ON/OFF   :
+K
  RESET TYPE   :
+R
  QUANTIZE     :
+Q
  PLAYLISTFILE :
*Pfile
  ICON LOADING :
+I
  MELODY       :
*Mc
  DRUMMAPFILE  :
*Ufile
  SOFT MUTING  :
+O
  COUNT        :
+C
  STATUS FIELD :
*Astatus
  FADE STEPS   :
*Fsteps
  LYRICTYPE    :
+J
```

```
LIST PLAYING:
+Y
PATCHCHANNEL:
*Nc1=c2,...
VOLUMECHANGE:
*Vc:X,...
MESSEAGEREXX :
*Lfile
FILENAMEREXX:
*Efile
FILETYPEREXX:
*Hfile
FORCERESET   :
+T
DIRBUFFER    :
+B
SETTINGS     :
*0file
```

1.8 STARTING FROM WORKBENCH

DOUBLE CLICK MidiPlay icon or MIDI file icon (= project icon ↔
with
default tool MidiPlay) or SELECT MIDI file icon and SHIFT DOUBLE CLICK
MidiPlay icon. All selected MIDI files are added to the playlist.
The tooltypes used in the program (default tool) icon are read.
Options in the configuration file should be in the tooltype format.

Tooltypes (=configuration options):

```
HELP MESSAGE

FILEINFO=ON

TEXTINFO=ON

DONOTPLAY=ON

TEMPO=VALUE

DATAINFO=ON

MUTE=C1,C2...

SOLO=C

TRANSPOSE=VALUE

DRUMCHANNEL=C

NOPC=ON

NOPB=ON
```

LOOP=ON

PRIORITY=VALUE

SEQ=ON

MERGE=ON

EXT=ON

CIA=ON

WINDOW=ON

SKIP=VALUE

NODRUMPC=ON

NOMAPCHANNELS=C1,C2...

BUFDEF=ON

START=ON

EXIT=ON

MIDISTART=ON

MAPFILE=file

MAP=ON

NRESET=ON

QUANTIZE=ON

PLAYLIST=file

ICON=ON

MELODY=c

DRUMMAPFILE=file

SOFT=ON

COUNT=ON

MFSTATUS=status

FADE=steps

LYRICTYPE=ON

PLAY=ON

PATCHCHANNEL=c1=c2,...

VOLUMECHANGE=c:X,...

MESSAGEREXX=file

FILENAMEREXX=file

FILETYPEREXX=file

FORCERESET=ON

DIRBUFFER=ON

CHANNELBYTE=VALUE

BANKCHANGETYPE=VALUE

NOSYSEX=ON

CHANNELPC=c:MAPINFO,...

KEYBOARD=file

DEFAULTKEYBOARD

PATTERN=pattern

SETTINGS=file

PORTNAME=port

MIDICLOCK=ON

INTROPLAY=ON

INTROTIME=time

REQTOOLS=ON

CHANNELMAP=structure

PUBSCREEN=screenname

LYRICFONT=size,font

LYRICCOLORS=text,highlight

MERGETRANSPOSE=VALUE

RELTRANSPOSE=ON

FILEDEFAULTS=file

MIXERPATH=file

APPICONFILE=file

APPICONNAME=name

APPICONPOS=x,y

FILEPATH=path

1.9 HELP

ToolType: -
ARexx : -
CLI : *?
keyboard: -
menu : -

Help message.

Default: -

1.10 MIDI FILE INFORMATION

ToolType: FILEINFO=<ON|OFF>
ARexx : SHOWFILEINFO <ON|OFF|TOGGLE>
CLI : <-|+>f
keyboard: -
menu : SETUP->SHOW OPTIONS->FILE INFO

Prints details of the MIDI file to the console window:
format, the number of the tracks and Steps/Quarter Note.
Will be shown only right after loading of a new file.
Same information is also available in About requester.

Default: OFF

1.11 META EVENT PRINTING

ToolType: TEXTINFO=<ON|OFF>
ARexx : SHOWMETA <ON|OFF|TOGGLE>
CLI : <-|+>x
keyboard: -
menu : SETUP->SHOW OPTIONS->META EVENTS

Prints all meta events in the file to the console window.
A meta event can express tempo, time signature, key
signature or a text event (copyright notice,
sequence/track name, instrument name, lyric etc.).

Default: OFF

See also:

COUNT

LYRIC WINDOW~

1.12 DO NOT SEND MIDI EVENTS

```
ToolType: DONOTPLAY=<ON|OFF>
ARexx : SETNOMIDIOUTPUT <ON|OFF|TOGGLE>
CLI : <-|+>n
keyboard: -
menu : SETUP->MIDI OPTIONS->NO MIDI OUTPUT
```

Prevents MidiPlay from sending events to MIDI port. Using this option you can print all the events without a long delay between events (delay is about 10ms).

Default: OFF

1.13 SET TEMPO

```
ToolType: TEMPO=<beats per measure>
ARexx : TEMPO <beats per measure>
CLI : *t<beats per measure>
keyboard: -
menu : -
Values : 40-240 or 0
```

Sets playing tempo to VALUE (BPM). Tempo change messages in the midi file are ignored. If value is 0, tempo is set to default value.

Default: 120 (and tempo messages not ignored)

See also:

TEMPODOWN

TEMPOUP

1.14 MIDI EVENT PRINTING

```
ToolType: DATAINFO=<ON|OFF>
ARexx : SHOWDATA <ON|OFF|TOGGLE>
CLI : <-|+>d
keyboard: -
menu : SETUP->SHOW OPTIONS->DATA EVENTS
```

Prints all real midi events in the file to the console window.

Default: OFF

1.15 MUTE CHANNELS

```
ToolType: MUTE=<sign><channel>,<channel>...
ARexx : MUTECHANNELS <sign><channel>,<channel>...
CLI : *m<sign><channel>,<channel>...
keyboard: -
menu : -
Values : 1-16, sign can be '-', '+' or nothing
```

Mutes all events on requested channels. If sign is '-', MUTE ON is set OFF, otherwise it is set ON. You can mute all selected channels by setting

```
MUTE ON
to ON.
```

If channel is 0, all channels will be unmuted. You can put MUTE=0 to your configuration file or to file defaults file.

Example: MUTE=1,5,9 mutes channels 1, 5 and 9.

Default: no muting

1.16 SOLO CHANNEL

```
ToolType: SOLO=<sign><channel>
ARexx : SOLOCHANNEL <sign><channel>
CLI : *s<sign><channel>
keyboard: -
menu : -
Values: 0-16
```

Sends events only on requested channel. Sets setting

```
SOLO ON
to ON when sign is not '-'. Otherwise it is set
off. By setting channel to 0 and SOLO ON on, you can
mute all channels at once.
```

Default: no soloing

1.17 TRANSPOSE

```
ToolType: TRANSPOSE=<value>
ARexx : TRANSPOSE <value>
CLI : *r<value>
keyboard: -
menu : -
Values : -12--+12
```

Transposes all note on/off messages by value (see also

```
DRUMCHANNEL
option).
```

Default: 0

1.18 SETS DRUM CHANNEL

```
ToolType: DRUMCHANNEL=<channel>
ARexx : DRUMCHANNEL <channel>
CLI : *u<value>
keyboard: -
menu : -
Values: 0-16
```

Sets drum channel. If this option is given, no transpose will occur on this channel.

Default: 10

1.19 PROGRAM CHANGE FILTER

```
ToolType: NOPC=<ON|OFF>
ARexx : SETNOPC <ON|OFF|TOGGLE>
CLI : <-|+>c
keyboard: -
menu : SETUP->FILTER OPTIONS->PROGRAM CHANGE
```

Program change messages are not passed if this option is set.

Default: OFF

See also:

```
NO DRUM PC
```

1.20 PITCH BENDER FILTER

```
ToolType: NOPB=<ON|OFF>
ARexx : SETNOPB <ON|OFF|TOGGLE>
CLI : <-|+>b
keyboard: -
menu : SETUP->FILTER OPTIONS->PITCH BENDER
```

Pitch bender messages are not passed if this option is set.

Default: OFF

1.21 LOOP

```
ToolType: LOOP=<ON|OFF>
ARexx : SETLOOP <ON|OFF|TOGGLE>
CLI : <-|+>l
keyboard: F6
menu : SETUP->PLAY OPTIONS->LOOP (AMIGA-O)
```

Loop. When the end of the song is reached, the song is played again.

Default: OFF

1.22 SETS PLAYER PRIORITY

```
ToolType: PRIORITY=<priority>
ARexx : PRIORITY <priority>
CLI : *p<priority>
keyboard: -
menu : -
Values : -128-+127
```

Sets the player priority.

Default : 30

1.23 SEQUENTIAL PLAYING

```
ToolType: SEQ=<ON|OFF>
ARexx : SETSEQUENCE <ON|OFF|TOGGLE>
CLI : <-|+>q
keyboard: -
menu : SETUP->PLAY OPTIONS->SEQUENTIAL
```

Plays tracks (in format 1 files) in sequence.

Default: OFF

1.24 MIDI MERGE

```

ToolType: MERGE=<ON|OFF>
ARexx : SETMERGE <ON|OFF|TOGGLE>
CLI : <-|+>g
keyboard: F4
menu : SETUP->MIDI OPTIONS->MERGE

```

Merges MIDI input to MIDI output (actually, it routes the MidiIn port to the MidiOut port).

Default: OFF

1.25 EXTERNAL SYNC

```

ToolType: EXT=<ON|OFF>
ARexx : SETSYNC <ON|OFF|TOGGLE>
CLI : <-|+>e
keyboard: -
menu : SETUP->MIDI OPTIONS->EXTERNAL SYNC

```

Receives clock signals from an external source and synchronizes to it. Tempo can be controlled in real time from the source. The program waits a Start or Continue message from that source before starting. MidiPlay can be stopped by sending a Stop message.

If this option is not given, MidiPlay transmits clock signals and Start and Stop messages unless option

MIDICLOCK is unset. Thus an external source can be synchronized to MidiPlay.

If

CIA timer is selected, it will be used in internal sync mode. Otherwise all timing will be done through timer.device.

Default: OFF (internal sync)

See also:

MIDI START

1.26 CIA TIMER

```

ToolType: CIA=<ON|OFF>
ARexx : SETCIA <ON|OFF|TOGGLE>

```

```
CLI : <-|+>i
keyboard: -
menu : SETUP->MISC OPTIONS->CIA TIMER
```

If this option is set and internal sync is used, MidiPlay will use CIA timer. Otherwise timer.device will be used. Use of CIA timer may improve timing.

Default: OFF

1.27 SHOWS MAIN WINDOW

```
ToolType: WINDOW=<ON|OFF>
ARexx : SETWINDOW <ON|OFF|TOGGLE>
CLI : <-|+>w
keyboard: -
menu : -
```

Shows the main window. ARexx command affects only window updating. If -w is used: there is no keyboard or menu available. The given file(s) will be played unless option -S is used. After all files have been played and if +X option is set, the program exits. Otherwise all files are played again. If loop is used, the first file will be played again. You can stop the program by pressing CTRL-C or CTRL-D (if started from CLI or started from Workbench and text window is open) or by using ARexx QUIT command.

Default: ON

1.28 SKIPS EMPTY MEASURES

```
ToolType: SKIP=<measures>
ARexx : SKIP <measures>
CLI : *k<measures>
keyboard: -
menu : -
```

Values : 1-

If this option is set, MidiPlay will skip all eventless measures if their number is bigger than given value.

Default: 20

1.29 NO PROGRAM CHANGE ON DRUM CHANNEL

```
ToolType: NODRUMPC=<ON|OFF>
ARexx : SETNODRUMPC <ON|OFF|TOGGLE>
CLI : <-|+>o
keyboard: -
menu : SETUP->FILTER OPTIONS->NO DRUM PC
```

If this option is set, MidiPlay will ignore all program change messages on the drum channel.

Default: OFF

See also:

NO PC

1.30 NO MAPPING ON CHANNELS

```
ToolType: NOMAPCHANNELS=<channel>,<channel>...
ARexx : NOMAPCHANNELS <channel>,<channel>...
CLI : *h<channel>,<channel>...
keyboard: -
menu : -
Values : 1-16
```

If this option is used, program change mapping won't apply to given channels. This option doesn't affect drum mapping. If channel is 0, NOMAPCHANNELS is reset to the default value.

Example: NOMAPCHANNELS=11,12 means that mapping won't apply to channels 11 and 12.

Default: mapping on all channels

1.31 BUFFER AS DEFAULT SELECTION

```
ToolType: BUFDEF=<ON|OFF>
ARexx : SETBUFDEFAULT <ON|OFF|TOGGLE>
CLI : <-|+>a
keyboard: -
menu : SETUP->MISC OPTIONS->BUFFER SELECT
```

When this options is used, default selection for the selection requester will be buffer. This means that if you select some number from the playlist, it won't be played until the current file has been played or you press CURSOR DOWN key. Default selection can always be overridden by pressing the PLAY gadget in the selection window.

Default: OFF

See also:

SELECTION WINDOW

1.32 IMMEDIATE START AFTER LOADING

```
ToolType: START=<ON|OFF>
ARexx : SETSTART <ON|OFF|TOGGLE>
CLI : <-|+>S
keyboard: F5
menu : SETUP->PLAY OPTIONS->START AFTER LOAD (AMIGA-T)
```

Playing starts immediately after the MIDI file has been loaded. If not set, the program will wait until you press PLAY or CONT gadgets, space bar or TAB key or send corresponding ARexx message.

Default: ON

1.33 EXIT AFTER PLAYING

```
ToolType: EXIT=<ON|OFF>
ARexx : SETEXIT <ON|OFF|TOGGLE>
CLI : <-|+>X
keyboard: -
menu : SETUP->PLAY OPTIONS->EXIT AFTER PLAY (AMIGA-X)
```

The program exits after the current file has been played. If MidiPlay has been started without a window and this option is set, the program will exit after all files in the playlist have been played.

Default: OFF

1.34 WAITS EXTERNAL START

```
ToolType: MIDISTART=<ON|OFF>
ARexx : SETMIDISTART <ON|OFF|TOGGLE>
CLI : <-|+>D
keyboard: -
menu : SETUP->MIDI OPTIONS->MIDI START
```

If external sync is used, the program won't start playing until it receives START or CONT message from the external source.

Default: ON

See also:

EXTERNAL SYNC

1.35 MAPPING

```
ToolType: MAP=<ON|OFF>
ARexx : SETMAP <ON|OFF|TOGGLE>
CLI : <-|+>K
keyboard: -
menu : SETUP->FILTER OPTIONS->MAP
```

If this option is set, program change messages will be mapped according to the map file or individual channel maps.

If this option is not set, the map file won't be used even if it were loaded.

Default: OFF (no mapping)

1.36 PROGRAM CHANGE MAPFILE

```
ToolType: MAPFILE=<filename>
ARexx : LOADMAP <filename>
CLI : *G<filename>
keyboard: -
menu : -
```

The program uses file for mapping the program change messages. See

content of mapfile
. If
MAP
is set off,
mapping won't be used.

Default: no mapfile

See also:

CHANNEL MAPS

1.37 NORMAL RESET

```
ToolType: NRESET=<ON|OFF>
ARexx : SETRESET <ON|OFF|TOGGLE>
CLI : <-|+>R
keyboard: -
menu : SETUP->RESET OPTIONS->NORMAL RESET
```

The program doesn't use GM reset message. It sends normal reset instead.

See also:

```
SENDRESET
```

```
FORCERESSET
```

```
Default: OFF (international)
```

1.38 QUANTIZE

```
ToolType: QUANTIZE=<ON|OFF>
ARexx : SETQUANTIZE <ON|OFF|TOGGLE>
CLI : <-|+>Q
keyboard: -
menu : SETUP->PLAY OPTIONS->QUANTIZE
```

The program quantizes all event times to the nearest multiple of "steps/measure" / 24.

Default: OFF

1.39 PLAYLIST FILE

```
ToolType: PLAYLIST=<filename>
ARexx : LOADPLAYLIST <filename>
CLI : *P<filename>
keyboard: -
menu : PROJECT->OPEN PLAYLIST (with requester)
```

The program loads file as a playlist.
See

```
File control
```

Default: no playlist

1.40 READS TOOLTYPES

```
ToolType: ICON=<ON|OFF>
ARexx : SETICON <ON|OFF|TOGGLE>
```

```
CLI : <-|+>I
keyboard: -
menu : SETUP->PLAY OPTIONS->LOAD .INFO
```

If this option is set and a new MIDI file is loaded, the command values are read from a configuration file for that file (if it exists). This file should be named to <file>.cfg unless it has an extension, in that case, the extension shouldn't be in the name of the configuration file. For example, if you have a MIDI file called mysong, the configuration file should be mysong.cfg, but if you have a file called mysong.mid, the configuration file should be mysong.cfg. Be careful with dots, "st.elmo" should have "st.cfg" as a configuration file! If the configuration file is not found, MidiPlay tries to read tooltypes from the file's icon (if there is an icon).

Before loading of the the tooltypes, MidiPlay tries to read tooltypes in the file defaults file (defined with

FILEDEFAULTS

command) and set those default file values.

Using this file you can decide which values should always be changed even when there is no icon for a file. You could use e.g. TEMPO=0 to set tempo to default value or MUTE=0 to unmute all channels.

Transposing in older versions of MidiPlay (< v2.60) was a little bit different. Value in file defaults file (def_midisplay icon) was absolute value and the value in a file icon was relative to that absolute value. You can get the same effect in this version by using transposeup or transposedown commands, but if you don't like to change all of your files, you can set ON the

RELTRANPOSE

option.

For example, if for some reason you want all files to be played in lower key preserving the values you have earlier given for files, put suitable negative transpose value to your file defaults file.

Default: OFF

1.41 MELODY CHANNEL

```
ToolType: MELODY=<sign><channel>
ARexx : MELODY <sign><channel>
CLI : *M<sign><channel>
keyboard: -
menu : -
Values : 0-16 (channel), sign can be '-', '+' or nothing
```

This sets the melody channel to given value and that channel is muted. If channel is negative (i.e. sign is '-'), the melody channel won't be muted at the beginning.

Default: 0

See also:

MELODY CHANNEL

MUTE MELODY

MELODY VOLUME DOWN

MELODY VOLUME UP

1.42 DRUM MAPFILE

```
ToolType: DRUMMAPFILE=<filename>
ARexx : LOADDRUMMAP <filename>
CLI : *U<filename>
keyboard: -
menu : -
```

The program uses file for mapping the note on/off messages on the drum channel. If MAP is set off, the map file will be loaded but it won't be used.

Default: no drum mapfile

See also:

MAPFILE

1.43 SOFT MUTING

```
ToolType: SOFT=<ON|OFF>
ARexx : SETSOFT <ON|OFF|TOGGLE>
CLI : <-|+>O
keyboard: -
menu : SETUP->MIDI OPTIONS->MUTE ONLY NOTES
```

If this option is used, all messages except note on/off will be passed on muted channels. In this case, the program uses two different routes between MidiPlay and MidiOut. The first one passes only note on/off messages and the other one all other messages. If unset, no messages will be passed.

Default: OFF

1.44 COUNT MEASURES AND COLLECTS TEXT EVENTS

```
ToolType: COUNT=<ON|OFF>
ARexx : SETCOUNT <ON|OFF|TOGGLE>
CLI : <-|+>C
keyboard: -
menu : SETUP->PLAY OPTIONS->COUNT
```

If this option is used, the number of the measures in a MIDI file and the duration of the file in seconds will be counted after the file has been loaded. This may take couple of seconds, but then you are able to see e.g. how many seconds are remain during playing.

During the counting operation also lyric events are collected for lyric output, track names are collected for the track name window and song name and copyright events are set to the corresponding fields in the main window. If this option is not set when the file is loaded, features mentioned above are not available.

Default: OFF

See also:

TRACK NAME WINDOW

LYRIC WINDOW

1.45 MEASURE FIELD STATUS

```
ToolType: MFSTATUS=<status>
ARexx : SETSTATUSFIELD <status>
CLI : *A<status>
keyboard: -
menu : -
Value : 0-3
```

Status tells what is the state of the measure field in the beginning. Status can be:

- 0 = none
- 1 = measure
- 2 = time
- 3 = remain

Default: 0

See also:

STATUS FIELD UP

1.46 FADE

```
ToolType: FADE=<steps>
ARexx : FADE <steps>
CLI : *F<steps>
keyboard: -
menu : -
Values : 0 -
```

Steps defines how many steps (tenth of seconds) fade out requires when playing is stopped. When volumes have been faded to zero, all notes off message is sent. Not every instrument reacts correctly to volume message (CC 7). This means that some sounds keep playing until notes off message is sent. 0 means that no fading will be done, pressing STOP gadget or SPACE BAR will stop playing but all notes keep playing until user sends all notes off message by pressing STOP gadget again or ESC.

Default: 0

1.47 LYRIC TYPE

```
ToolType: LYRICTYPE=<ON|OFF>
ARexx : SETLYRICTYPE <ON|OFF|TOGGLE>
CLI : <-|+>J
keyboard: -
menu : -
```

This option is not available since 2.51beta.

Lyric type defines what text events will be seen in the lyric window. If this is set ON, only normal lyric events (FF 5) will be seen, otherwise normal text events (FF 1) will be seen.

Default: ON

1.48 SETS PLAYLIST PLAYING

```
ToolType: PLAY=<ON|OFF>
ARexx : SETPLAYLIST <ON|OFF|TOGGLE>
CLI : <-|+>Y
keyboard: F7
menu : SETUP->PLAY OPTIONS->AUTO LOAD NEXT (AMIGA-Y)
```

When this option is set, MidiPlay will load a new file from the playlist after the last one has been played. MidiPlay plays first the number 1 in the beginning of the program. Exception: if playlist has been given and other files are given on the command line or other file icons are selected, other files will be added to the end of the

playlist and playing is started from the first file after the given playlist files.

The current file number played is shown in the window title after the file name, like file :1/Amount. If the file name has an extension, it won't be shown. Amount means the number of files in the playlist.

The current number can be change back to 1 using FIRST or SELECT. If you want that a new file is loaded but playing is not started, unset START AFTER LOAD.

Default: ON

See also:

```
FILE CONTROL
START AFTER LOAD
SELECTION WINDOW
PLAYLIST WINDOW
```

1.49 CHANNEL PATCHING

```
ToolType: PATCHCHANNEL=<channel=channel>,<channel=channel>...
ARexx : PATCHCHANNEL <channel=channel>,<channel=channel>...
CLI : *N<channel=channel>,<channel=channel>...
keyboard: -
menu : -
Values : 0-16
```

If this option is used, every message originally meant for the first channel will be sent on the second channel instead of the first one. If the first channel is 0 and the second channel is greater than 0, all channel will be set to the value of the second channel. If both are zeros, channels are set to normal values, i.e. 1=1,2=2,3=3...

See also:

```
PATCH CHANNEL WINDOW
Default: no patch
```

1.50 VOLUMECHANGE

```
ToolType: VOLUMECHANGE=<channel:change>,<channel:change>...
ARexx : SETVOLUMECHANGE <channel:change>,<channel:change>...
```

```

CLI : *V<channel:change>,<channel:change>...
keyboard: -
menu : -
Values : 0-16 for channel, -128-+127 for change

```

If this option is used, change will be added to every volume message (CC 7) on channel. change can also be negative. If change is 0, nothing will be added. If channel is 0, volumechange on all channels (1-16) will be set to the change value.

Default: 0 (all channels)

See also:

VOLUMEDOWN

VOLUMEUP

1.51 DEFINES SCRIPT FOR MESSAGES

```

ToolType: MESSAGEREXX=<filename>
ARexx : MESSAGEREXX <filename>
CLI : *L<filename>
keyboard: -
menu : -

```

This option replaces old NOMON option. filename is an ARexx script, that will be called every time a message is shown on the screen. The script gets the contents of the message as an argument. See also

Example 1

.

Default: no script

1.52 DEFINES SCRIPT FOR FILENAMES

```

ToolType: FILENAMEREXX=<filename>
ARexx : FILENAMEREXX <filename>
CLI : *E<filename>
keyboard: -
menu : -

```

This options replaces old SPEAK option. filename is an ARexx script, that will be called after a new file has been loaded. The script gets the following argument: Drive,Path,File,Extension (commas included). You can use this script e.g. to speak the filename or to view a text file associated with this MIDI file. See also

Example 2

Default: no script

1.53 DEFINES SCRIPT FOR NON-MIDI FILES

```

ToolType: FILETYPEREXX=<filename>
ARexx : FILETYPEREXX <filename>
CLI : *H<filename>
keyboard: -
menu : -

```

This option defines an ARexx script, which will be called when the current file is not a MIDI file. The main purpose of this script could be unpacking archived files. The script gets the following argument: Drive,Path,File,Extension (commas included). See also

Example 3

Default: no script

1.54 FORCES RESET

```

ToolType: FORCERESET=<ON|OFF>
ARexx : SETFORCERESET <ON|OFF|TOGGLE>
CLI : <-|+>T
keyboard: -
menu : SETUP->RESET OPTIONS->FORCE RESET

```

If this option is set, every channel will be reset in spite of the mute/PC/PB settings. If this option is not set, current settings are taken into account.

Default: ON

See also:

RESET

SEND RESET

1.55 REMEMBERS LAST DIRECTORY

```

ToolType: DIRBUFFER=<ON|OFF>
ARexx : SETDIRBUFFER <ON|OFF|TOGGLE>
CLI : <-|+>B
keyboard: -

```

menu : SETUP->MISC OPTIONS->DIRECTORY BUFFER (AMIGA-B)

If this option is set, the MIDI file load requester will remember the contents of the current directory. If unset, the requester will scan the current directory everytime it is opened.

This option available only for
ReqTools
requesters.

Default: ON

1.56 CHANNELBYTE FOR TONE SYSEX

ToolType: CHANNELBYTE=<bytenumber>
ARexx : CHANNELBYTE <bytenumber>
CLI : -
keyboard: -
menu : -
Values : 3 - ?

This option is used in external tone files. If some program change is mapped to an external tone file, which consists of one system exclusive message, bytenumber tells which byte in the sysex message is replaced with program change's channel.

Default: 3 (This is a device ID on most synths)

1.57 TYPE OF BANK CHANGE MESSAGE

ToolType: BANKCHANGETYPE=<typenumber>
ARexx : BANKCHANGETYPE <typenumber>
CLI : -
keyboard: -
menu : -
Values : 0,1,2 or 32

This option defines the bank change message which is sent when a program change is mapped to a bank change + program change combination. There seems to be conflicting bank changes on different synths. Some require at least CC 0, but others CC 32. Value 0 means that the MSB byte is 0, if the value is 32, the MSB is 32, e.g. if bank change 3 is going to be sent and typenumber is 0, CC 0 3 CC 32 0 will be sent, but if typenumber is 32, CC 0 0 CC 32 3 will be sent.

If your synth doesn't recognize these bank changes, but they can be defined somehow with a sysex message, contact me with a correct format so that I can add it to MidiPlay.

Value 1 is for Roland D-10 and value 2 for Yamaha FB01.

Default: 0

1.58 SYSEX MESSAGES IN FILES NOT SENT

```
ToolType: NOSYSEX=<ON|OFF>
ARexx : SETNOSYSEX <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->FILTER OPTIONS->SYSEX
```

If this option is set, all system exclusive messages in MIDI files are filtered out.

Default : OFF

1.59 PROGRAM CHANGE FOR CURRENT FILE

```
ToolType: CHANNELPC=<channel:matype>,<channel:matype>...
ARexx : CHANNELPC <channel:matype>,<channel:matype>...
CLI : -
keyboard: -
Values : 1-16 for channel
```

This tooltype should be used only with individual MIDI files and it shouldn't be used in the configuration file or in the default MidiPlay icon.

When CHANNELPC is used, MidiPlay sends defined program change message right before the playing is started. All program change messages encountered in the current MIDI file are ignored. All CHANNELPC settings are cleared when a MIDI file is loaded.

Note: If a MIDI file includes a SYSEX reset, all channelpc settings are lost.

matype can be:

-value value is mapped according to map file,
value can be between 0-127

value value is sent as a program change,
value can be between 0-127

Bbank:value bank change and a program change are sent,
bank and value can be between 0-127

Ttonefilename message from tonefile is sent, correct
channel is set according to
CHANNELBYTE

,

only 128 tonefiles can be in memory in any time

C clears CHANNELPC setting on channel

For example, if CHANNELPC=4:-10 is used, program change defined in the current mapfile on row 11 is sent on channel 4. If CHANNELPC=3:B2:20 is used, bank change 3 and program change 20 are sent on channel 3.

See also

MAP FILE FORMAT

.

Default : No forced program changes

1.60 CONNECTS AN EXTERNAL MIXER

ToolType: -
 ARexx : CONNECTMIXER <ON|OFF|TOGGLE>
 CLI : -
 keyboard: -
 menu : WINDOWS->CONNECT MIXER

If this command is set, an external MIDI mixer will be connected to MidiPlay. If the mixer isn't running, it will be started. By unsetting this option the connection is removed, but the mixer is not quitted. If the mixer has been started from MidiPlay, it will be closed at exit. Currently only Mixer included in MidiPlay ToolKit is supported.

When the mixer is running, control, note and program change messages are sent to the mixer. When the mixer is started from MidiPlay or the connection is made to already running mixer, all current volumes are sent to the mixer.

Default : Not connected.

1.61 TOGGLES SELECTED CHANNEL

ToolType: -
 ARexx : MUTETOGGLE <channel>
 CLI : -
 keyboard: <1>|<2>|<3>|<4>|<5>|<6>|<7>|<8>|<9>|<0>|<q>|<w>|<e>|<r>|<t>|<y>
 menu : -
 Values : 1-16

If this command is sent, the state of the channel is changed, i.e. if it is muted, it will be unmuted and if it is unmuted, it will be muted.

See also:

MUTE CHANNELS

MUTE ON

1.62 TOGGLES SELECTED CHANNEL

```
ToolType: -
ARexx : STATUSFIELDUP
CLI : -
keyboard: u
menu : -
```

If this command is sent, the state of the measure field is changed to the next state. See

MEASURE FIELD STATUS

.

1.63 ACTIVATES THE NEXT OPEN WINDOW

```
ToolType: -
ARexx : ACTIVATENEXTWIN
CLI : -
keyboard: i
menu : -
```

If this command is sent, the next open window will be activated.

1.64 ACTIVATES THE DRUM CHANNEL GADGET

```
ToolType: -
ARexx : ACTIVATEDRUMGAD
CLI : -
Keyboard: d
menu : -
```

If this command is sent and preferences window is open, the

drum channel
gadget will be activated.

1.65 ACTIVATES THE SOLO CHANNEL GADGET

```
ToolType: -
ARexx : ACTIVATESOLOGAD
CLI : -
Keyboard: s
menu : -
```

If this command is sent and preferences window is open, the

```
    solo channel
    gadget will be activated.
```

1.66 ACTIVATES THE MELODY CHANNEL GADGET

```
ToolType: -
ARexx : ACTIVATEMELODYGAD
CLI : -
Keyboard: l
menu : -
```

If this command is sent and preferences window is open, the

```
    melody channel
    gadget will be activated.
```

1.67 VOLUME CHANGE OF THE MELODY CHANNEL DOWN

```
ToolType: -
ARexx : MELODYVOLUMEDOWN <amount>
CLI : -
Keyboard: <LSHIFT>+<F8>
menu : -
```

If this command is sent, the volume change of the melody channel will be decreased by amount.

See also:

```
MELODY CHANNEL
```

```
MELODY VOLUME UP
```

1.68 VOLUME CHANGE OF THE MELODY CHANNEL UP

```
ToolType: -
ARexx : MELODYVOLUMEUP <amount>
CLI : -
Keyboard: <RSHIFT>+<F8>
menu : -
```

If this command is sent, the volume change of the melody channel will be increased by amount.

See also:

MELODY CHANNEL

MELODY VOLUME DOWN

1.69 LOADS KEYBOARD SETTINGS

```
ToolType: KEYBOARD <filename>
ARexx : LOADKEYBOARD <filename>
CLI : -
keyboard: -
menu : -
```

If this command is sent, keyboard settings will be loaded from filename. Keyboard files can be edited by using KeyboardEditor (included in MidiPlay ToolKit).

Keyboard files map only raw key codes so custom keyboard files should preserve the correct keys underlined in the user interface, e.g. letters a, s, d, l, m should have correct functions if you want to keep the interface consistent. Only default keyboard hard coded in MidiPlay maps these keys correctly on different local keymaps.

Default: Default keyboard

1.70 RETURNS DEFAULT KEYBOARD

```
ToolType: DEFAULTKEYBOARD
ARexx : DEFAULTKEYBOARD
CLI : -
keyboard: -
menu : -
```

If this command is sent, the default keyboard is set back.

See also:

LOAD KEYBOARD

1.71 DEFINES PATTERN USED IN MIDI FILE REQUESTER

```
ToolType: PATTERN=<pattern>
ARexx : -
CLI : -
keyboard: -
menu : -
```

This setting defines the pattern which will be used in the MIDI file requester. This pattern will be used every time. Clearing the file requester's pattern gadget won't clear this setting i.e. it will be shown every time the requester is opened.

Default: No pattern

1.72 DEFINES CONFIGURATION FILENAME

```
ToolType: SETTINGS=<filename>
ARexx : -
CLI : *0
keyboard: -
menu : -
```

SETTINGS defines the name of the configuration file which will be loaded before any other tooltypes or CLI options will be read. You should put SETTINGS tooltype into MidiPlay (tool) icon. This tooltype is useless in project icons.

```
Examples: SETTINGS=S:midisplay.config (ToolType)
          +0S:midisplay.config (CLI)
```

Default: midisplay.prefs or ENV:MidiPlay/midisplay.prefs

1.73 DEFINES THE NAME OF MIDIPLAY'S AREXX PORT

```
ToolType: PORTNAME=<arexx portname>
ARexx : -
CLI : -
keyboard: -
menu : -
```

PORTNAME defines the name of the MidiPlay's ARexx port. Normally, MidiPlay_rexx is used, but if you don't like it, you can change it via this tooltype. This tooltype can be put into MidiPlay icon or in the SETTINGS file. The name

of the port cannot be changed after it has been created.

Example : PORTNAME=myport

Default : MidiPlay_rexx

See also:

MIDIPLAY AREXX COMMANDS

1.74 ICONIFIES OR DEICONIFIES MIDIPLAY

ToolType: -
ARexx : ICONIFY <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : PROJECT->ICONIFY (AMIGA-F)

If ICONIFY ON command is used, all MidiPlay windows will be closed and only a small icon will be shown on Workbench Screen. This icon is an AppIcon. Every icon (MIDI file!) dragged into this icon causes the corresponding file to be added to the playlist (but it is not automatically played). If ICONIFY OFF command is sent when MidiPlay is iconified, MidiPlay will be returned to the original window positions before it was iconified (this is the same action as double clicking on the AppIcon).

Example : ICONIFY ON (Iconifies MidiPlay)

Default : -

1.75 MIDI START/CLOCK/STOP MESSAGES WILL BE SENT

ToolType: MIDICLOCK=<ON|OFF>
ARexx : SETMIDICLOCK <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->MIDI OPTIONS->SEND MIDI CLOCK

If this option is set, and internal sync is used, START message will be sent when a file is started, STOP message will be sent when a file has been played and CLOCK messages will be sent (24 times per quarter-note).

Default : ON

1.76 SENDS NOTEON MESSAGE

```

ToolType: -
ARexx : NOTEON <channel:note>,<channel:note>,...
CLI : -
keyboard: -
menu : -
Values : channel 0-16
        note = letter (sharp) octave e.g. C3 or F#4

```

This command sends NOTEON (0x90) MIDI command on the defined channel. If channel is 0, melody channel will be used. Velocity 100 is always used. Remember always send the corresponding NOTEOFF command!

```

Example : NOTEON 0:C3,0:D#3,0:G3 (sends C minor chord on
        melody channel)

```

```

Default : -

```

1.77 SENDS NOTEOFF MESSAGE

```

ToolType: -
ARexx : NOTEOFF <channel:note>,<channel:note>,...
CLI : -
keyboard: -
menu : -

```

This command sends NOTEOFF (0x80) MIDI command on the defined channel. If channel is 0, melody channel will be used.

```

Example : NOTEOFF 0:C3,0:D#3,0:G3 (turns off C minor chord on
        melody channel)

```

```

Default : -

```

1.78 PROGRAM CHANGE MAPS FOR DIFFERENT CHANNELS

```

ToolType: CHANNELMAP=<channel>,<mapstruct>,<mapstruct>,...
ARexx : CHANNELMAP <channel>,<mapstruct>,<mapstruct>,...
CLI : -
keyboard: -
menu : -
values : channel 0-17 (0 = all channels, 17 = global map
        table = MAPFILE)

```

With this options it is possible to have different maps for different channels. The exact map structure is shown below:

```

mapstruct::= <C> the global map will be used on the given
        channel

```

mapstruct::= <F><filename> the map file will be used on the given channel. You cannot use channel 0 (i.e. all channels) in this case

mapstruct::= <A><mapformat> all PCs will be set to the mapformat value on the given channel

mapstruct::= <pcnumber><:><mapformat> PC with number will be set to the mapformat value. You cannot refer to all channels with this structure (use global map instead)

mapformat::= <T><tonefile> (use of this can cause troubles!)

mapformat::= <banknumber><:><pcnumber>

mapformat::= <pcnumber>

Examples:

CHANNELMAP=1,A20 (all pcs will be mapped to 20 on channel 1)

CHANNELMAP=3,10;B1:10 (pc 10 will be mapped to bank 1 pc 10 on channel 3)

CHANNELMAP=2,C (all pcs on channel 2 will be mapped according to global map i.e. the normal map file which can also be referred as channel 17, e.g. CHANNELMAP=17,4:5 will map pc 4 to 5 on channels which are set as mapchannels and there are no local map i.e. structure C has been used)

Remember to set MAP option and clear first all channels with CHANNELMAP=0,C in your config file (if you don't clear, local maps will be used on all channels and every pc will be mapped to 0!). You can put these settings one after another like CHANNELMAP=2,C,1:2,2:3,3:4 i.e. every row can refer to only one channel or all channels with channel 0.)

See also:

MAPPING ON

MAP FILE

DRUMMAP FILE

NO MAP CHANNELS

1.79 INTROPLAY

```
ToolType: INTROPLAY=<ON|OFF>
ARexx : SETINTROPLAY <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->PLAY OPTIONS->INTRO PLAY
```

When this option is set, the current file is played only a short time defined by
INTROTIME
option.

Default : OFF

1.80 INTROTIME

```
ToolType: INTROTIME=<seconds>
ARexx : INTROPLAYTIME <seconds>
CLI : -
keyboard: -
menu : -
```

INTROTIME value defines how long the current file will be played when
INTROPLAY
option is set.

Default : 10 seconds

1.81 REQTOOLS REQUESTERS WILL BE USED INSTEAD OF ASL

```
ToolType: REQTOOLS=<ON|OFF>
ARexx : SETREQTOOLS <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->MISC OPTIONS->USE REQTOOLS
```

When this option is not set, normal ASL requester or easy requesters will be used. When this option is set, requesters in the reqtools.library will be used if this library has been found.

Default : OFF

1.82 MIDIPLAY OPENED ON AN EXISTING PUBLIC SCREEN

```

ToolType: PUBSCREEN=<public_screenname>
ARexx : -
CLI : -
keyboard: -
menu : -

```

If PUBSCREEN has been defined in the configuration file or in the MidiPlay's tool types, MidiPlay will be opened on the given public screen. If this public screen doesn't exist, MidiPlay will be opened on the default public screen (Workbench).

```
Default : NULL
```

1.83 FONT FOR THE LYRIC WINDOW

```

ToolType: LYRICFONT=<fontsize>,<fontname>
ARexx : LYRICFONT <fontsize>,<fontname>
CLI : -
keyboard: -
menu : -

```

With this command you can define the font to be used in the

```

    lyric window
    .

```

The ".font" ending must be included. Use of non-proportional font is strongly recommended.

```
Example : LYRICFONT=11,topaz.font
```

```
Default : System default text
```

1.84 COLORS FOR THE LYRICS

```

ToolType: LYRICCOLORS=<text color>,<highlight color>
ARexx : LYRICCOLORS <text color>,<highlight color>
CLI : -
keyboard: -
menu : -
value : 0-7 for colors in the current color palette

```

You can select colors for the lyric window with this command. The first number tells the text colors which will be used to print the text. The second number tells the color which will be used to highlight the word to be sung. The colors can be changed via palette preferences in your prefs directory.

See also

lyric window
Default : 1,2

1.85 SELECT BY NUMBER FROM PLAYLIST

ToolType: -
ARexx : SELECTNUMBER <number>
CLI : -
keyboard: -
menu : -
values : number 1 - <entries in the playlist>

With this option, an entry can be selected from the playlist via ARexx without opening the selection window.

Default : -

See also:

SELECTION WINDOW

1.86 SELECTION TO BUFFER

ToolType: -
ARexx : SELECTBUFFER <number>
CLI : -
keyboard: -
menu : -
values : number 1 - <entries in the playlist>

With this option, an entry can be selected from the playlist via ARexx without opening the selection window. The selected number will be put into the
buffer
.

Default : -

See also:

SELECTION WINDOW

SELECT NUMBER

1.87 PLAYLIST LOCKED

```
ToolType: -
ARexx : SETLOCK <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->MISC OPTIONS->LOCK PLAYLIST
```

If this option is set, you can add/edit/delete entries in the playlist via

```
    playlist window
    without affecting playing
(unless the deleted entry is played).
```

If this option is not set, a click in the playlist will cause the selected entry to be played, add command will cause the added file (the first one if more than one) to be played, edit command will cause the edited filename to be loaded and delete will cause the currently played entry to be deleted from the playlist and the next one in the playlist to be played if there are any left).

Default : OFF

1.88 ALL NOTES OFF MESSAGE SENT

```
ToolType: -
ARexx : NOTESOFF
CLI : -
keyboard: ESC
menu : FUNCTIONS->ALL NOTES OFF
```

All notes on all channels are turned off.

Default : -

1.89 PLAY COMMAND

```
ToolType: -
ARexx : PLAY
CLI : -
keyboard: TAB
menu : -
```

Playing is started from the beginning of the current file.

1.90 STOP COMMAND

```
ToolType: -
ARexx : STOP or STOP TOGGLE
CLI : -
keyboard: SPACE (= STOP TOGGLE)
menu : -
```

If STOP command without the TOGGLE keyword is sent, playing will be stopped. If the TOGGLE keyword is used and the playing was already stopped, it will be continued.

1.91 CONTINUE PLAYING COMMAND

```
ToolType: -
ARexx : CONT
CLI : -
keyboard: SPACE (continue when playing stopped)
menu : -
```

1.92 LOADS FILE

```
ToolType: -
ARexx : LOAD <filename>
CLI : -
keyboard: -
menu : PROJECT->ADD FILE (AMIGA-L)
```

The given file will be loaded and played, and it will be added to the playlist.

See also:

ADD

FILE

1.93 LOADS FILE, NOT ADDED TO THE PLAYLIST

```
ToolType: -
ARexx : FILE <filename>
CLI : -
keyboard: -
menu : -
```

The given file will be loaded and played, but it won't be added to the playlist.

See also:

```
ADD
LOAD
```

1.94 ADDS FILE TO THE PLAYLIST

```
ToolType: -
ARexx : ADD <filename>
CLI : -
keyboard: -
menu : -
```

The given file will be added to the playlist, but it won't be loaded or played.

See also:

```
LOAD
FILE
```

1.95 TRANSPOSE UP

```
ToolType: TRANSPOSEUP=<value>
ARexx : TRANSPOSEUP <value>
CLI : -
keyboard: a
menu : -
```

Adds value to the transpose value.

See also

```
TRANSPOSE
```

1.96 TRANSPOSE DOWN

```
ToolType: TRANSPOSEDOWN=<value>
ARexx : TRANSPOSEDOWN <value>
CLI : -
keyboard: <A>|<z>
menu : -
```

Subtracts value from the transpose value.

See also

TRANSCOPE

1.97 TEMPO UP

```
ToolType: TEMPOUP=<value>
ARexx : TEMPOUP <value>
CLI : -
keyboard: m
menu : -
```

Adds value to the tempo value.

See also

TEMPO

1.98 TEMPO DOWN

```
ToolType: TEMPODOWN=<value>
ARexx : TEMPODOWN <value>
CLI : -
keyboard: <M>|<n>
menu : -
```

Subtracts value from the tempo value.

See also

TEMPO

1.99 QUITS MIDIPLAY

```
ToolType: -
ARexx : QUIT
CLI : -
keyboard: F1
menu : PROJECT->QUIT (AMIGA-Q)
```

MidiPlay is quitted. All windows are closed and reset message is sent.

1.100 NEW PLAYLIST

```
ToolType: -
ARexx : REMPLAYLIST
CLI : -
keyboard: -
menu : PROJECT->NEW PLAYLIST
```

Old playlist is removed.

1.101 LOADS PLAYLIST

```
ToolType: -
ARexx : LOADPLAYLIST <filename>
CLI : -
keyboard: -
menu : PROJECT->OPEN PLAYLIST (AMIGA-A) (the filename is
      fetched via a file requester)
```

Removes the old playlist and loads the given playlist.

1.102 FILTER FOR TEMPO CHANGES

```
ToolType: -
ARexx : SETTEMPOCHANGE <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->FILTER->TEMPO CHANGE
```

When this option is set, all tempo changes in MIDI files will be ignored.

Default : OFF

See also:

SET TEMPO

TEMPOUP

TEMPODOWN

1.103 OPENS/CLOSE THE LYRIC WINDOW

```
ToolType: -
ARexx : SHOWLYRIC <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : WINDOWS->LYRIC
```

Opens (ON) or closes (OFF) the lyric window
. If the

keyword TOGGLE is used, the lyric window is opened, if it's currently closed and closed if it's currently opened.

1.104 OPENS (CLOSES) THE SELECTION WINDOW

```
ToolType: -
ARexx : SELECT <number>
CLI : -
keyboard: F10
menu : FUNCTIONS->SELECT (AMIGA-C) (= SELECT 0)
values : number 0-2
```

If the number is 0, the selection window will be opened and the number gadget is activated. If the number is 1, the file name gadget is activated. If the given number is 0 and the selection window is open, it will be closed without any action.

See the
selection window
description.

1.105 MUTES/UNMUTES THE MELODY CHANNEL

```
ToolType: -
ARexx : SETMELODY <ON|OFF|TOGGLE>
CLI : -
keyboard: F8
menu : -
```

This options can be used to mute/unmute the melody channel.
See also

```
MELODY CHANNEL
and
MUTE ON
descriptions.
```

Default : OFF

1.106 SELECTS NEXT NUMBER FROM BUFFER

```
ToolType: -
ARexx : BUFFERNEXT
CLI : -
keyboard: CURSOR DOWN
menu : FUNCTIONS->BUFFER NEXT
```

A number is picked from the
buffer
(if the buffer is not
empty) and the corresponding MIDI file is loaded.

1.107 CLEARS THE BUFFER

```

ToolType: -
ARexx : BUFFERCLEAR
CLI : -
keyboard: DEL
menu : FUNCTIONS->BUFFER CLEAR

```

The
 buffer
 is cleared.

See also:

BUFFER NEXT

1.108 FAST FORWARD

```

ToolType: -
ARexx : FORWARD <value>
CLI : -
keyboard: f
menu : -

```

Playing will be set to the fast forward state. The speed will be $\text{value} \times 1.25$ times faster than the current tempo. FORWARD 0 should be used to set the playing back to the original tempo.

Default : 0

1.109 VOLUME UP

```

ToolType: VOLUMEUP=<channel:amount>,<channel:amount>...
ARexx : VOLUMEUP <channel:amount>,<channel:amount>...
CLI : -
keyboard: <RSHIFT> + <1>|<2>|<3>|<4>|<5>|<6>|<7>|<8>|<9>|<0>|
         <q>|<w>|<e>|<r>|<t>|<y>
menu : -
values : channel 0-16, amount -128 - +127

```

Adds amount to the value of volumechange on the given channel (if channel = 0, all volumechanges will be affected).

See also:

VOLUMECHANGE

VOLUME DOWN

1.110 VOLUME DOWN

```
ToolType: VOLUMEDOWN=<channel:amount>,<channel:amount>...
ARexx : VOLUMEDOWN <channel:amount>,<channel:amount>...
CLI : -
keyboard: <LSHIFT> + <1>|<2>|<3>|<4>|<5>|<6>|<7>|<8>|<9>|<0>|
          <q>|<w>|<e>|<r>|<t>|<y>
menu : -
values : channel 0-16, amount -128 - +127
```

Subtracts amount from the value of volumechange on the given channel (if channel = 0, all volumechanges will be affected).

See also:

VOLUMECHANGE

VOLUME DOWN

1.111 SYSTEM EXCLUSIVE COMMAND

```
ToolType: -
ARexx : SYSEX <hexstring>
CLI : -
keyboard: -
menu : -
```

System exclusive message defined by hexstring is sent. The hexadecimal string can be e.g. 41 20 16 00 or 41,20,16,00. Do not include starting F0 nor ending F7. They will be added automatically.

1.112 SEND NORMAL MIDI DATA

```
ToolType: -
ARexx : SENDMIDIDATA <hexstring>
CLI : -
keyboard: -
menu : -
```

MIDI data message defined by hexstring is sent. MIDI data bytes should be separated by spaces or commas. For example, The hexadecimal string can be 90 40 40 or 90,40,40. One MIDI message is started by a command number and it is followed by 1 or 2 bytes. Command number is

always hex 80 or greater and data bytes following it must be smaller than hex 80. You can send more than one MIDI message in the same string, e.g. 90 40 40 90 43 40 90 47 40.

A MIDI message can be (n = channel number 0-F):

```
8n ss ss NOTEOFF
9n ss ss NOTEON
An ss ss POLYPRESSURE
Bn ss ss CONTROL CHANGE
Cn ss PROGRAM CHANGE
Dn ss CHANNEL PRESSURE
En ss ss PITCH BENDER
```

or system message between F1 and FF. Use SYSEX command for messages starting with F0 (F7/EOX messages are not allowed).

1.113 MIDI MERGE TRANPOSE VALUE

```
ToolType: MERGETRANSPOSE=<value>
ARexx : MERGETRANSPOSE <value>
CLI : -
keyboard: -
menu : -
Values : -128 - 127
```

Transposes all note on/off messages from MIDI in by value. These messages are then sent transposed to MIDI out if option

```
MERGE
is set. Events on
DRUMCHANNEL
are not
```

transposed.

Default: 0

1.114 RANDOM PLAY

```
ToolType: -
ARexx : SETSHUFFLE <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : SETUP->PLAY OPTIONS->RANDOM PLAY (AMIGA-H)
```

When this option is set, the next file in the playlist

will be selected at random. Setting this option ON will cause an immediate selection. Because the corresponding menu option only toggles between states, you must first set it OFF and then set it back ON to make a new random selection.

If you have the MidiPlay ToolKit you can make your own key for the immediate selection by putting SETSHUFFLE ON to the description of the key combination. If you don't have the Keyboard Editor from the ToolKit you can use programs which have ability to assign ARExx scripts to hotkeys and use then the corresponding ARExx command (i.e. ADDRESS MidiPlay_rexx; SETSHUFFLE ON).

Default : OFF

1.115 PLAY THE FIRST FILE IN THE PLAYLIST

```
ToolType: -
ARExx : FIRST
CLI : -
keyboard: CURSOR UP
menu : FUNCTIONS->FIRST (AMIGA-1)
```

The first file in the playlist is played.

1.116 PLAY THE NEXT FILE IN THE PLAYLIST

```
ToolType: -
ARExx : NEXT
CLI : -
keyboard: CURSOR RIGHT
menu : FUNCTIONS->NEXT (AMIGA->)
```

The next file in the playlist is played.

1.117 PLAY THE PREVIOUS FILE IN THE PLAYLIST

```
ToolType: -
ARExx : PREVIOUS
CLI : -
keyboard: CURSOR LEFT
menu : FUNCTIONS->PREVIOUS (AMIGA-<)
```

The previous file in the playlist is played.

1.118 RELATIVE TRANSPOSE WITH FILES

```
ToolType: RELTRANSPOSE=<ON|OFF>
ARexx : SETRELTRANSPOSE <ON|OFF|TOGGLE>
CLI : -
keyboard: -
menu : -
```

When this option is ON, all transpose values in files' configuration will be relative, i.e. they are added to the value of the current transpose value. This option is for compatibility with previous versions of MidiPlay. When this option is OFF, all transpose values will be treated as such. You can get the same effect as with older versions of MidiPlay by using commands TRANSPOSEUP=<value> or TRANSPOSEDOWN=<value>.

See also option

```
ICON
Default : OFF
```

1.119 DEFAULTS FOR FILES

```
ToolType: FILEDEFAULTS=<filename>
ARexx : FILEDEFAULTS <filename>
CLI : -
keyboard: -
menu : -
```

This command replaces the def_midisplay.info file used with older versions (< v2.60) of MidiPlay. The user can now tell the name of the file where the default values for files are. This file has same the format as the MidiPlay configuration file. With this file, you can tell which values should be set (to their defaults) before the next file is loaded. These values could be e.g. volumechanges or transpose.

See also option

```
ICON
Default : no file defaults
```

1.120 DEFAULT DIRECTORY FOR MIDI FILE

```
ToolType: FILEPATH=<filepath>
ARexx : FILEPATH <filepath>
CLI : -
keyboard: -
menu : -
```

You can give the default directory for MIDI files with this command. If you use this command in your

configuration file or MidiPlay's icon, MidiPlay's file requester will be opened directly in the defined directory.

Example: FILEPATH=sys:midifiles/

Default : no default directory

1.121 DIRECTORY PATH FOR THE MIXER

```
ToolType: MIXERPATH=<filename>
ARexx : MIXERPATH <filename>
CLI : -
keyboard: -
menu : -
```

You can tell the directory path of the mixer with this command. For example, if you have your mixer in a sys:midi_app directory, you should use command MIXERPATH="sys:midi_app/Mixer". If you rename the mixer, the changed name should be given in this command.

Default : "Mixer"

1.122 USER DEFINABLE APPLICATION ICON

```
ToolType: APPICONFILE=<filename>
ARexx : APPICONFILE <filename>
CLI : -
keyboard: -
menu : -
```

You can give your own icon to be used in MidiPlay when it is iconified. filename tells the directory path of the icon file, but the ".info" ending must be excluded. For example, if you want to use the icon of the ENV:sys/def_tool.info file, you should give APPICONFILE="ENV:sys/def_tool". You can get the built-in version by giving APPICONFILE without arguments.

See also:

ICONIFY

APPICONNAME

APPICONPOS

Default : Built-in icon

1.123 NAME SHOWN UNDER APPICON

```

                ToolType: APPICONNAME=<name>
ARexx : APPICONNAME <name>
CLI : -
keyboard: -
menu : -

```

You can give the name of the appicon shown below the appicon with this command. For example, APPICONNAME="MidiPlay app".

See also:

```

                ICONIFY

                APPICONFILE

                APPICONPOS
                Default : -

```

1.124 APPICON POSITION

```

                ToolType: APPICONPOS=<x,y>
ARexx : APPICONPOS <x,y>
CLI : -
keyboard: -
menu : -

```

With this command, you can give the position for the appicon on the Workbench screen. x tells the horizontal position and y tells the vertical position of the icon. NOTE: if this position is occupied by another icon, Workbench finds a free position for the appicon.

An example: APPICONPOS=<50,100>

See also:

```

                ICONIFY

                APPICONNAME

                APPICONFILE
                Default : no icon positions

```

1.125 QUERY FOR SETTINGS OF MIDIPLAY

```

                ToolType: -
ARexx : STATUS <variablename>
CLI : -
keyboard: -
menu : -

```

This ARexx command gives the value of the requested

setting (variablename). Most of the MidiPlay's settings can be requested. Available variables are listed below. If RC variable in ARexx script is 0, command was recognized and the variablename was known ARexx command name (this variable is not necessarily available). The result is in RESULT variable. Otherwise, RC has value 1.

Example script

Available variables:

- 1) These following variables can have values 0 (off) or 1 (on). More information in the command documentation part of this guide.

BUFFERCLEAR
result 1 if buffer empty

ICONIFY
result 1 if iconified

PLAY
result 1 if playing

SETBUFDEFAULT

SETCIA

SETCOUNT

SETDIRBUFFER

SETEXIT

SETFORCERESET

SETICON

SETINTROPLAY

SETLOCK

SETLOOP

SETMUTE

SETMERGE

SETMIDISTART

SETNODRUMPC

SETNOMIDIOUTPUT

SETMAP

```
SETMELODY
    result 1 if melody channel really muted

SETMIDICLOCK

SETNODRUMPC

SETNOPB

SETNOPC

SETNOSYSEX

SETPLAYLIST

SETQUANTIZE

SETRELTRANSPOSE

SETREQTOOLS

SETRESET

SETSEQUENCE

SETSHUFFLE

SETSOFT

SETSOLO

SETSTART

SETSYNC

SETTEMPOCHANGE

SETWINDOW
    result 1 if the main window is open

SHOWFILEINFO

SHOWMETA

SHOWDATA

SHOWLYRIC
    result 1 if the lyric window is open

STOP
    result 1 if stopped, otherwise 0
```

2) The following variables return string values:

```
LOADPLAYLIST
    the name of the last opened/saved playlist
```

LOADMAP
the name of the current map file

LOADDRUMMAP
the name of the current drum map file

FILENAMEREXX
the name of the filename script

FILETYPEREXX
the name of the filetype script

MESSAGEREXX
the name of the message script

FILEDEFAULTS
the name of the file defaults file

LYRICFONT
the name of the current lyric font

LYRICCOLORS
a string telling normal and highlight colors for lyrics, e.g. "1,2"

APPICONPOS
a string telling appicon position in format x-coord,y-coord, e.g. "50,50"

MUTECHANNELS
a string of muted channels separated by commas

NOMAPCHANNELS
a string of not mapped channels separated by commas

APPICONNAME
the name shown under appicon

MIXERPATH
directory path of the mixer

3) The following variables return numerical values:

BANKCHANGETYPE

BUFFERNEXT
the playlist number of the next file in the buffer

CHANNELBYTE

DRUMCHANNEL

FADE

```
FORWARD

INTROPLAYTIME

MERGETRANSPOSE

MELODY
  melody channel

PRIORITY

SETSTATUSFIELD

SKIP

SOLOCHANNEL

TEMPO

TRANSPPOSE
```

1.126 Example status script

```
/* request muted channels, these channels are muted
   only if muteon is set also */

options results
address 'MidiPlay_rexx'

'status mutechannels'
if rc == 0 then
  say 'Muted channels:' result
'status setmute'
if rc == 0 then
  if result == 1 then
    say 'muteon is on'
  else
    say 'muteon is off'
```

1.127 WINDOWS

```
MAIN WINDOW

PREFERENCES WINDOW
  open via menu:
SETUP
->
MISC OPTIONS

TRACK NAME WINDOW
  open via menu:
```

WINDOWS

LYRIC WINDOW

open via menu:

WINDOWS

PLAYLIST WINDOW

open via menu:

WINDOWS

SELECTION WINDOW

open via menu:

FUNCTIONS

1.128 MAIN WINDOW

MENUS

GADGETS

KEYBOARD

1.129 MENUS

PROJECT

FUNCTIONS

SETUP

WINDOWS

1.130 PROJECT

NEW PLAYLIST

OPEN PLAYLIST... (AMIGA A)

A file requester is opened and using this requester you can load a new playlist. Be careful what you select, playlist is not verified in any ways, so MidiPlay will probably crash if you load a file which is not a playlist.

See also

LOADPLAYLIST

command.

SAVE PLAYLIST... (AMIGA S)

A file requester is opened and using this requester you can save the current playlist to a file.

ADD FILE... (AMIGA L)

A file requester is opened. Using this requester you can add files to the end of the playlist. The first (in the alphabetic order) will be played.

See also commands:

ADD

LOAD

FILE

REMOVE FILE

This one does nothing at this moment.

ABOUT (AMIGA ?)

Information about the program and the current file will is showed.

ICONIFY

(AMIGA F)

SAVE ENVIRONMENT

All window positions and open/closed/zoomed information are saved to ENV: MidiPlay/midiplay.win and ENVARC: MidiPlay/midiplay.win files. ENV: MidiPlay/midiplay.win file will be loaded when MidiPlay is started and windows will be positioned correctly unless WINDOW=OFF or -w option is used. In this case, windows won't be opened.

QUIT

(AMIGA Q, F1)

1.131 FUNCTIONS

SEND RESET

(AMIGA R)

ALL NOTES OFF

FIRST

(AMIGA 1, CURSOR UP)

PREVIOUS

(AMIGA <, CURSOR LEFT)

NEXT

(AMIGA >, CURSOR RIGHT)

SELECT...
(AMIGA C, F10)

BUFFER NEXT
(CURSOR DOWN)

BUFFER CLEAR
(DEL)

1.132 SETUP MENU

MIDI OPTIONS

RESET OPTIONS

PLAY OPTIONS

FILTER OPTIONS

SHOW OPTIONS

MISC OPTIONS

1.133 MIDI OPTIONS

EXTERNAL SYNC
(AMIGA E)

SEND MIDI CLOCK

MUTE ON
(AMIGA M, F2)

SOLO ON
(AMIGA N, F3)

MUTE ONLY NOTES

MERGE
(F4)

NO MIDI OUTPUT

MIDI START
SET CHANNELS... (AMIGA P)

Opens the

preferences window
PATCH CHANNELS...

Opens the

Patch channels window

1.134 RESET OPTIONS

NORMAL RESET

FORCE RESET

1.135 PLAY OPTIONS

LOAD .INFO

COUNT

LOOP
(AMIGA O, F6)

SEQUENTIAL

QUANTIZE

START AFTER LOAD
(AMIGA T, F5)

EXIT AFTER PLAY
(AMIGA X)

AUTO LOAD NEXT
(AMIGA Y)

RANDOM PLAY
(AMIGA H)

INTRO PLAY

1.136 FILTER OPTIONS

PROGRAM CHANGE

NO DRUM PC

PITCH BENDER

TEMPO CHANGE

SYSEX

MAP

1.137 SHOW OPTIONS

FILE INFO

META EVENTS

DATA EVENTS

1.138 MISC OPTIONS

DIRECTORY BUFFER
(AMIGA B)

BUFFER SELECT

CIA TIMER

LOCK PLAYLIST

USE REQTOOLS

1.139 MUTE ON

ToolType: - (see
MUTE
)

ARexx : SETMUTE <ON|OFF|TOGGLE>

CLI : -

keyboard: F2

menu : SETUP->MIDI OPTIONS->MUTE ON (AMIGA-M)

If this option is set, all the selected mute channels will be muted. If SOLO ON is set at the same time, only the solo channel will be played. If this option is not set, all channels will be unmuted (unless SOLO ON is set).

See also

MUTECHANNELS
Default : OFF

1.140 SOLO ON

```
ToolType: - (see
SOLO
)
ARexx : SETSOLO <ON|OFF|TOGGLE>
CLI : -
keyboard: F3
menu : SETUP->MIDI OPTIONS->SOLO ON (AMIGA-N)
```

If this option is set, only the selected solo channel will be played.

See

```
SOLOCHANNEL
Default : OFF
```

1.141 RESET

```
ToolType: -
ARexx : SENDRESET
CLI : -
menu : FUNCTIONS->SEND RESET
```

If normal reset is not set, the program will send GM reset message and ALL NOTES OFF messages before the first MIDI event and also before quitting. If normal reset is set, the program will send the following messages before loading a new MIDI file and also before quitting:

- PROGRAM CHANGE 0
- CONTROLLERS 0 and 32 to 0
- MAIN VOLUME to 100
- PAN to 64
- EXTDEPTH to 64
- EXPRESSION to 127
- RPN COURSETUNE to 64, RPN FINETUNE to 64, RPN PBSSENS to 2
- SUSTAIN to 0
- PORTA to 0
- MODWHEEL to 0
- MAX to 0
- RESET CONTROLLERS
- ALL NOTES OFF

See also

```
FORCERESET
```

1.142 BUFFER

There can be 9 numbers in the buffer at most. You can select the next number from the buffer by pressing CURSOR DOWN key or by selecting BUFFER NEXT from the menu. If there is a number in the buffer when the current file has been played, it will be loaded. The next file in the buffer is shown in the screen title of the main window.

You can add entries to the buffer via the
selection window
or via
SELECTNUMBER
command.

1.143 SELECTION WINDOW

MENUS

You can select the number of the MIDI file in the playlist through the selection requester. Just write the right number. You can also write the name of the file (or parts of it) into the file gadget. You can move to another gadget by pressing TAB. If the given file (or file number) is not found, the screen flashes. The requester is not closed until cancelled (by pressing ESC or Right Amiga + C when one of the edit gadgets is selected, or C when none of the edit gadget is selected) or legal file is selected. When you are editing, you can press F10 which clears the currently selected field (just like CONTROL+X).

You can select the number edit gadget by pressing letter n and the file name edit gadget by pressing letter f when none of the gadgets are selected. Possible file numbers are shown in the window title bar.

If you press PLAY gadget (or F8, P when an edit gadget is not selected, or Right Amiga + P, when one of the edit gadget is selected) the selected file will be loaded immediately. If you press BUFFER gadget (or F9, B when an edit gadget is not selected, or Right Amiga + B, when one of the edit gadget is selected), the number will be put to a buffer.

Note: the name gadget is case sensitive!

See also:

BUFFER

SELECTBUFFER

SELECT

1.144 MAPFILE

Normal map converts every program change message to the maptypes defined in the mapfile. Row number tells which program change message (0-127) is mapped. To use the mapping, MAP option must be set. Maptypes can be:

a value between 0-127, e.g. 120

Bbank:value this is a bank change+program change combination
e.g. B2:20

Ttonefilename program change is mapped to a message in the external tonefile. Correct

CHANNELBYTE

should be

set. Tone files are first read to memory. Number of tone files in memory is limited to 128, including those in map files and used with

CHANNELPC

. Old ones are replaced

with new ones. External tone files are pure system exclusive dump files, not MIDI files.

The drum map converts every note on/off message on the drum channel to the number specified in the drum map file (Numbers run from 0 to 127, one per row). To use the mapping, MAP option must be set.

See

CHANNELMAP

for making maps for

individual channels

See an

example map file

including the names of the General

MIDI instruments

1.145 EXAMPLE MAP FILE

Here is an example map file including the names of the General MIDI instruments. These names and the numbers in front of the names are not needed for mapping. Only the first number/maptype is needed. If tonefilename is given, those instrument names must not be given on that line! In this example, # mark is for commenting purpose only and means that the line ends there and the rest of the line should be removed if used in a real situation.

-----The example file starts after this line-----

```
0 1 :Acoustic Grand Piano
1 2 :Bright Acoustig Piano
```

2 3 :Electric Grand Piano
3 4 :Honky Tonk Piano
B1:4 5 :Electric Piano 1 #electric piano is selected from bank 1
5 6 :Electric Piano 2
6 7 :Harpsichord
Tsounds:clavi # 8 :Clavi clavi sound is loaded from a sysex file
8 9 :Celesta
8 10 :Glocken spiel # glocken spiel is played as a celesta
10 11 :Music Box
11 12 :Vibraphone
12 13 :Marimba
13 14 :Xylophones
14 15 :Tubular Bells
15 16 :Dulcimer
16 17 :Drawbar Organ
17 18 :Percussive Organ
18 19 :Rock Organ
19 20 :Church Organ
20 21 :Reed Organ
21 22 :Accordion
22 23 :Harmonica
23 24 :Tango Accordion
24 25 :Acoustic Guitar (N)
25 26 :Acoustic Guitar (S)
26 27 :Electric Guitar 1
27 28 :Electric Guitar 2
28 29 :Electric Guitar 3
29 30 :Overdriven Guitar
30 31 :Distortion Guitar
31 32 :Guitar Harmonica
32 33 :Acoustic Bass
33 34 :Electric Bass (finger)
34 35 :Electric Bass (pick)
35 36 :Fretless Bass
36 37 :Slap Bass 1
37 38 :Slap Bass 2
38 39 :Synth Bass 1
39 40 :Synth Bass 2
40 41 :Violin
41 42 :Viola
42 43 :Cello
43 44 :Contrabass
44 45 :Tremolo Strings
45 46 :Pizzicato Strings
46 47 :Orchestral Strings
47 48 :Timbani
48 49 :String Ensemble 1
49 50 :String Ensemble 2
50 51 :Synth Strings 1
51 52 :Synth Strings 2
52 53 :Choir Aahs
53 54 :Voice Oohs
54 55 :Synth Voice
55 56 :Orchestral Hit
56 57 :Trumpet
57 58 :Trombone
58 59 :Tuba

59	60	:Muted Trumpet
60	61	:French Horn
61	62	:Brass Section
62	63	:Synth Brass 1
63	64	:Synth Brass 2
64	65	:Soprano Sax
65	66	:Alto Sax
66	67	:Tenor Sax
67	68	:Baritone Sax
68	69	:Oboe
69	70	:English Horn
70	71	:Bassoon
71	72	:Clarinet
72	73	:Piccolo
73	74	:Flute
74	75	:Recorder
75	76	:Pan Flute
76	77	:Blown Bottle
77	78	:Shakuhachi
78	79	:Whistle
79	80	:Ocarina
80	81	:Lead 1 (Square)
81	82	:Lead 2 (Sawtooth)
82	83	:Lead 3 (Calliope)
83	84	:Lead 4 (Chiff)
84	85	:Lead 5 (Charang)
85	86	:Lead 6 (Voice)
86	87	:Lead 7 (Fifths)
87	88	:Lead 8 (Bass+lead)
88	89	:Pad 1 (New Age)
89	90	:Pad 2 (Warm)
90	91	:Pad 3 (Polysynth)
91	92	:Pad 4 (Choir)
92	93	:Pad 5 (Bowed)
93	94	:Pad 6 (Metallic)
94	95	:Pad 7 (Halo)
95	96	:Pad 8 (Sweep)
96	97	:FX 1 (Rain)
97	98	:FX 2 (Soundtrack)
98	99	:FX 3 (Crystal)
99	100	:FX 4 (Atmosphere)
100	101	:FX 5 (Brightness)
101	102	:FX 6 (Goblins)
102	103	:FX 7 (Echoes)
103	104	:FX 8 (SCI-FI)
104	105	:Sitar
105	106	:Banjo
106	107	:Shamisen
107	108	:Koto
108	109	:Kalimba
109	110	:Bagpipe
110	111	:Fiddle
111	112	:Shanai
112	113	:Tinkle Bell
113	114	:Agogo
114	115	:Steel Drums
115	116	:Wood Block

```
116 117:Taiko Drum
117 118:Melodic Tom
118 119:Synth Drum
119 120:Reverse Cymbal
120 121:Guitar Fret Noise
121 122:Breath Noise
122 123:Seashore
123 124:Bird Tweet
124 125:Telephone Ring
125 126:Helicopter
126 127:Applause
127 128:Gunshot
```

1.146 FILE CONTROL

Playlist file is a file that includes names and directories of the MIDI files, like `dh0:music/MID/file.mid`, one per row. The first file is number 1, the second is number 2 and so on unless you begin a row with '#' mark, then the rest of the row is ignored and it is not included in the playlist.

You can tell the program the name of the playlist file in the configuration file, through tooltypes or CLI arguments. All files given as arguments or icons are added to the end of the playlist. Playlist can be changed using open/save/new playlist in the project menu.

1.147 WINDOWS

You can open/close one of the following window by selecting its name. If a window is open, there is a checkmark in front of its name.

TRACK NAME WINDOW

LYRIC WINDOW

PLAYLIST WINDOW

CONNECT MIXER

1.148 GADGETS

The NAME field shows the name of the sequence (format 0) or the ↵ name of the first track (format 1) which usually is the name of the sequence.

The copyright "(C)" field shows the copyright event if there is one.

Tempo can be changed using the TEMPO SLIDER. Tempo will not be changed until the left mouse button is released.

The status field shows the state of the player. END and WAITING means that the program is waiting the user to do something. WAIT EXT means that the program is waiting START or CONT message from an external source. PLAY means that the program is currently playing a MIDI file. You can see the number of the current measure, time elapsed or time remain by changing the state of the cycle gadget. CHECKING means that the file was not recognized as a MIDI file, but FILETYPEREXX command is run (if defined). You can use this command e.g. to unpack packed MIDI files.

The CYCLE gadget is used to change the type of the information in the status field left from the cycle gadget. If NONE is selected, only WAIT, PLAY or END messages will be seen. If MEASURES is selected, the current measure number will be seen. If COUNT option is set, also the number of the measures in the file will be shown. If TIME is selected, elapsed time will be shown. If REMAIN is selected and COUNT option is set, remaining time will be shown.

The ZOOM gadget toggles between the whole window and a window with only title bar at the top-left corner of the screen.

The following graphical symbols are used:

```
[BLACK SQUARE]   :
                  STOP
                  |>   :
                  PLAY
                  >    :
                  CONT
                  >>   :
                  FAST FORWARD
                  |<<   :
                  PREVIOUS
                  >>|   :
                  NEXT
                  V [ARROW DOWN] : BUFFER NEXT (meaning of this may change in
future)
```

By pressing STOP gadget you can stop playing. If Fade is set to 0 and you press this gadget twice, ALL NOTES OFF messages will be sent on all channels. If Fade is greater than 0, playing fades out (it will take $100\text{ms} \times \text{Fade}$), and ALL NOTES OFF messages will be sent after fading.

By pressing PLAY gadget you can start playing from the start of the file.

By pressing CONT gadget you can continue playing from the current position in the MIDI file.

By pressing FAST FORWARD gadget down you can make MidiPlay play faster. When you release the gadget, normal tempo is returned. FF speed used for this gadget is 75% faster than the original tempo.

By pressing PREV gadget you can load the previous file from the playlist.

By pressing NEXT gadget you can load the next file from the playlist.

1.149 KEYBOARD

Custom keyboard can be made with KeyboardEditor see
MidiPlay ToolKit

LSHIFT means that LEFT SHIFT key should be pressed at the same time as other key(s). The main use of LSHIFT key is to decrease
e.g. volumechange.

RSHIFT means that RIGHT SHIFT key should be pressed at the same time as other key(s). The main use of RSHIFT key is to increase
e.g. volumechange.

SPACE BAR depending on the state of the program, stops/continues playing

TAB starts playing from the start of the file

ESC sends ALL NOTES OFF messages on all channels

F1 quit

F2 MUTE ON ON/OFF

F3 SOLO ON ON/OFF

F4 merge ON/OFF

F5 start after load ON/OFF

F6 loop ON/OFF

F7 playlist play ON/OFF

F8 mutes/unmutes melody channel

LSHIFT+F8 decrease volumechange on the melody channel by one

RSHIFT+F8 increase volumechange on the melody channel by one

F10 select from playlist

CURSOR UP plays the first file from the playlist

CURSOR LEFT plays the previous file from the playlist

CURSOR RIGHT plays the next file from the playlist

CURSOR DOWN reads the next playlist number from the buffer

DEL clears the buffer

1,2,3,4,5,

6,7,8,9,0,

q,w,e,r,t,y use these keys to mute/unmute channels. These keys corresponds the channels 1-16.

LSHIFT+key decreases volumechange on this channel by one

RSHIFT+key increases volumechange on this channel by one

a transpose up

A,

z transpose down

m tempo up

M,

n tempo down

i activates the next open window
 f fast forward 75% faster than the original tempo,
 by making a custom keyboard file or using ARexx command,
 you can use any value greater than 0 for FORWARD command.
 FORWARD 0 returns the original tempo.

u changes the state of the status field cycle gadget
 l activates the melody channel gadget in the preferences
 window
 d activates the drum channel gadget in the preferences
 window
 s activates the solo channel gadget in the preferences
 window

1.150 PREFERENCES WINDOW

MENUS

KEYBOARD

NOTE: Menu shortcuts in the main window are NOT available when \leftrightarrow
 this
 window is active. All other keyboard shortcuts are available.

MUTE gadgets channels 1-16. If MUTE ON is set, checked channels will be muted.

The SOLO field tells what channel will be heard if SOLO ON is set. Pressing s selects this gadget.

The DRUM CHANNEL field tells what is the drum channel that won't be transposed. Pressing d selects this gadget.

The

MELODY CHANNEL
 field. Pressing l selects

this gadget.

MidiPlay transposes note on/note off messages according to the TRANSPOSE SLIDER. Messages on the drum channel will not be transposed.

NOTE: If any of the channel is muted and option SOFT is not set when playing is started, the program change, volume, pan and other control messages will NOT be transmitted. This means that instrument settings are not correct when those channels are unmuted.

1.151 MELODY CHANNEL

The melody channel can be muted/unmuted pressing F8 in the main window. For example the melody channel can be given using the tooltypes in the icon of the MIDI file (MELODY=4). The melody channel is mostly different

than in previous files. If you want to play or sing along the file, you may need to mute the melody channel.

1.152 PATCH CHANNELS WINDOW

MENUS

KEYBOARD

This window can be used to patch MIDI events on a MIDI channel to be played on another channel. The radio buttons on the left side of the window, under the "From" text, tell the channel which is patched to the channel shown on the right side of the window, under the "To" text. For example, if you want to patch channel 5 to channel 8, select first 5 on the left side and then select 8 on the right side.

See also

PATCHCHANNEL
command.

1.153 TRACK NAME WINDOW

MENUS

KEYBOARD

The listview gadget will show all track names defined as track name events in the current MIDI file. This listview is updated only right after loading the MIDI file and only if


COUNT

option is set on. You must reload the file, if you had COUNT set off.

1.154 PLAYLIST WINDOW

MENUS

KEYBOARD

Listview gadget in this window shows all the files in the  playlist.

Text gadget under the listview shows the name of the currently selected/played file.

Lock checkbox gadget is checked:

the text gadget shows the file currently loaded and you can select a new file to be played by clicking the name of

that file in the listview. If you press ADD gadget, a file requester opens and you can add files to the playlist. When you exit the requester without canceling, the selected file (or the first from the selected) will be loaded. All selected files will be added to the end of the playlist. If you press EDIT gadget, an edit file requester opens and you can edit the name of the selected file. When you exit this requester without canceling, the edited file will be loaded. If you press REMOVE gadget, the selected file will be removed from the playlist and the next one in the playlist will be loaded.

Lock checkbox gadget is not checked:

You can edit the playlist without affecting playing. Selection from the playlist causes that file to be removable/editable and the name will be shown under the listview gadget.

If you press ADD gadget, a file requester opens and you can add files to the playlist. When you exit the requester without canceling, all selected files will be added to the end of the playlist. If you press EDIT gadget, an edit file requester opens and you can edit the name and the path of the selected file. If you press REMOVE gadget, the selected file will be removed from the playlist. Playing will be stopped if the removed file was playing and the next one in the playlist will be loaded.

1.155 LYRIC WINDOW

MENUS

KEYBOARD

All lyrics will be shown in this window. The COUNT options should have been on when the current file was loaded, because during the counting operation, all lyrics are collected to a structure from which the word/syllable to be sung is selected during playing.

You can change the font of the lyric window with
LYRICFONT
command and the text colors with
LYRICCOLORS
command.

MIDI karaoke files

If the file is recognized as a MIDI karaoke file, MidiPlay tries to act according to the format of this file type. Depending on the window size, MidiPlay tries to show two paragraphs or the number of the rows that can be fitted into the window (in the case that one paragraph cannot be fitted into the window).

The current word/syllable is highlighted. When the current paragraph has been played, the window will scroll so that the next paragraph is shown at the top of the window. The scrolling will happen also, if the number of the rows on the screen has been exceeded. The highlighting will be used only on the top paragraph.

If you resize the window, the latest word/syllable will be shown at the top-left corner of the window. The window should be big enough to show at least 4 rows of text for the best performance.

If the window was closed and you open it, no lyric is shown in the window until the next lyric event in the file has been encountered.

Ordinary MIDI files

If the file is not recognized as a MIDI karaoke file and COUNT option has been used, all lyric events found in the file will be collected and shown in the window. All words that fit into the window are shown without any paragraphs (unless these lyric events use MIDI karaoke formatting commands). This may look ugly, but it will look much uglier if these events include new line characters (I can't help you in this case). I have tried to check at least the first character and if it is a new line character, this event will be ignored.

You can resize/open/close this window and it will act just like in the karaoke file case.

Note: if the lyric events have newline characters, lyrics won't be correctly shown. I've tried to remove the newline characters at the beginning of these events, but those in the middle of the event cannot be detected at this moment.

1.156 AREXX COMMANDS

MidiPlay opens one ARexx port called MidiPlay_rexx.

See STARTING MIDIPLAY section when MESSAGEREXX, FILETYPEREXX and FILENAMEREXX scripts will be called.

If MidiPlay doesn't recognize used keyword, it returns value 1, otherwise 0 is returned.

The following keywords can be used with some of the commands:

ON is same as 1 (option is set ON)
OFF is same as 0 (option is set OFF)
TOGGLE is same as -1 (option is set ON if OFF and OFF if ON)

The following commands are currently supported:

Playing commands:

```
PLAY

STOP

STOP TOGGLE

CONT

QUIT

TEMPOUP <value>

TEMPODOWN <value>

TEMPO <value>

SETTEMPOCHANGE <ON|OFF|TOGGLE>

FORWARD <value>

FADE <value>

SETLOOP <ON|OFF|TOGGLE>

SETSHUFFLE <ON|OFF|TOGGLE>

SETINTROPLAY <ON|OFF|TOGGLE>

INTROPLAYTIME <seconds>
  Playlist commands:

LOAD <filename>

FILE <filename>

ADD <filename>

REPLAYLIST

LOADPLAYLIST <filename>

SETSTART <ON|OFF|TOGGLE>

SETEXIT <ON|OFF|TOGGLE>

SETPLAYLIST <ON|OFF|TOGGLE>
```

```
SETLOCK <ON|OFF|TOGGLE>

FIRST

NEXT

PREVIOUS

SELECT <number>

SELECTNUMBER <number>
  Miscellaneous commands:

STATUS <variablename>

SETICON <ON|OFF|TOGGLE>

SETCOUNT <ON|OFF|TOGGLE>

BUFFERNEXT

BUFFERCLEAR

SETBUFDEFAULT <ON|OFF|TOGGLE>

SELECTBUFFER <number>

SETCIA <ON|OFF|TOGGLE>

SETSEQUENCE <ON|OFF|TOGGLE>

SETQUANTIZE <ON|OFF|TOGGLE>

PRIORITY <priority>

SKIP <measures>

FILEDEFAULTS <filename>

FILEPATH <path>

FILENAMEREXX <filename>

FILETYPEREXX <filename>

MESSAGEREXX <filename>

SETSTATUSFIELD <status>

STATUSFIELDUP

ACTIVATENEXTWIN

ACTIVATEDRUMGAD
```

```
ACTIVATESOLOGAD
ACTIVATEMELODYGAD
LOADKEYBOARD <filename>
DEFAULTKEYBOARD
SETDIRBUFFER <ON|OFF|TOGGLE>
SETWINDOW <ON|OFF|TOGGLE>
ICONIFY <ON|OFF|TOGGLE>
CONNECTMIXER <ON|OFF|TOGGLE>
SETREQTOOLS <ON|OFF|TOGGLE>
SETRELTRANPOSE <ON|OFF|TOGGLE>
MIXERPATH <filename>
APPICONFILE <filename>
APPICONNAME <name>
APPICONPOS <x,y>
  MIDI commands:

TRANPOSE <value>
TRANPOSEUP <value>
TRANPOSEDOWN <value>
MERGETRANPOSE <value>
SENDRESET
NOTESOFF
SYSEX <hexstring>
SENDMIDIDATA <hexstring>
NOTEON <channel:note>,<channel:note>...
NOTEOFF <channel:note>,<channel:note>...
SETSYNC <ON|OFF|TOGGLE>
MUTECHANNELS <sign><channel>,<channel>...
SOLOCHANNEL <sign><channel>
SETMUTE <ON|OFF|TOGGLE>
```

```
SETSOLO <ON|OFF|TOGGLE>
MUTETOGGLE <channel>
SETMERGE <ON|OFF|TOGGLE>
SETMIDISTART <ON|OFF|TOGGLE>
SETNOMIDIOUTPUT <ON|OFF|TOGGLE>
SETMIDICLOCK <ON|OFF|TOGGLE>
SETRESET <ON|OFF|TOGGLE>
SETFORCERESET <ON|OFF|TOGGLE>
SETNOPC <ON|OFF|TOGGLE>
SETNOPB <ON|OFF|TOGGLE>
SETNODRUMPC <ON|OFF|TOGGLE>
SETNOSYSEX <ON|OFF|TOGGLE>
SETSOFT <ON|OFF|TOGGLE>
SETMAP <ON|OFF|TOGGLE>
LOADMAP <mapfile>
LOADDRUMMAP <mapfile>
NOMAPCHANNELS <channel>,<channel>...
CHANNELMAP <channel>,<mapstruct>,<mapstruct>,...
CHANNELPC <channel:maptypes>,<channel:maptypes>...
CHANNELBYTE <value>
BANKCHANGETYPE <value>
DRUMCHANNEL <channel>
MELODY <sign><channel>
SETMELODY <ON|OFF|TOGGLE>
MELODYVOLUMEDOWN <amount>
MELODYVOLUMEUP <amount>
PATCHCHANNEL <channel=channel>,<channel=channel>...
SETVOLUMECHANGE <channel:amount>,<channel:amount>...
```

```
VOLUMEDOWN <channel:amount>,<channel:amount>...
```

```
VOLUMEUP <channel:amount>,<channel:amount>...
```

```
Print commands:
```

```
SHOWFILEINFO <ON|OFF|TOGGLE>
```

```
SHOWMETA <ON|OFF|TOGGLE>
```

```
SHOWDATA <ON|OFF|TOGGLE>
```

```
SHOWLYRIC <ON|OFF|TOGGLE>
```

```
SETLYRICTYPE <ON|OFF|TOGGLE>
```

```
LYRICFONT <size>,<font>
```

```
LYRICCOLORS <text>,<highlight>
```

1.157 OTHER INFORMATION

- When started from Workbench, midiplay opens its own console window for text input. That window is not visible unless any text is printed.
- Midiplay opens one midi source named "midiplay". Source is connected to MIDI library's resident node "MidiOut". MidiPlay opens also one MIDI destination named "midiplayin" which is connected to "MidiIn" when the player is playing and uses external synchronization.

1.158 TODO

- map file editor
- intelligent remapping
- Amiga's internal sounds played on midi channels

1.159 PROBLEMS AND LIMITATIONS

On small Amigas (500 & 600) Midiplay does not multitask very well. Files are played in time, but sometimes tempo is jerky. Some files have sometimes so many events in one measure that MidiPlay is not able to send all the events exactly right time on slow Amigas. You can improve timing by using CIA timer and/or using quantization.

If you change the time from the workbench preferences, MidiPlay hangs. If you don't want this to happen, use CIA timer instead.

MidiIn may lose some clock signals on slow Amigas and the midiplayer is

then out of sync. If this happens, use slower tempo and do not print anything on the screen.

Faulty MIDI files may cause problems. If you see "Can't read MIDI file" on the screen, loading was incomplete. You can still listen them, but you may get into troubles because of missing END OF TRACK messages etc.

Some enforcer hits at exit still to be fixed.

Limitations:

-SMPTE messages are ignored.

1.160 HISTORY

v1.0 29-Jun-93

first version, only DOS, format 0 files.

v1.1

23-Jul-93

first public release, WB and tooltypes, format 1 files.

v1.2 29-Aug-93

external/internal sync and midi merge added, SYSEX messages now transmitted correctly, too slow tempo problem fixed.

v1.2b 17-Oct-93

some corrections, smaller code

v2.0beta 20-Nov-93

added graphical user interface, transpose bug fixed, CIA timer not finished yet

v2.0 24-Jan-94

added: playlist, map file support, reset message, lyric and measure fields, melody muting, loading of MIDI file icons, quantization. Priority now set back to 0 when exiting. CIA timer works.

v2.0.1 27-Feb-94

fixed int. reset bug, added mapchannel and skip options

v2.01 28-Feb-94

added drum map and soft muting option.

v2.01a 06-Mar-94

added NODRUMPC option, added short delay after loading when NRESET is used.

v2.10beta 14-Mar-94

added own task for the playing routine: printing and using the right mouse button or opening windows/requesters won't stop playing, this means that printing e.g. data events are not necessarily in real time. Printed events look clearer. Fixed memory loss bug, also needs now less stack.

2-Apr-94

added BUFDEF (selection can be put to a buffer), COUNT (length of the file counted) and NOMON (sends arexx messages to Upd) options. Fixed "merge not set after loading" bug. Added different time information and a cycle gadget to select the information type. Many options are set to configuration values when the tool types of a file is loaded.

8-Apr-94

fixed one serious bug in data event printing.

15-Apr-94

changed the minibuffer to hold 9 numbers at maximum. Transpose values in tool types are now added to the configuration value. Options MFSTATUS and SPEAK added.

16-Apr-94

added keyboard shortcuts for tempo, transpose, channel muting, melody/drumchannel/solo channel selection, window activation. New option FADE.

17-Apr-94

all files are now added to the playlist. More than one filename can be given on the command line. More than one icon can be selected and they all will be added to the playlist. File control functions has been changed: MidiPlay doesn't load a new file unless play option is set (default is on), minibuffer is an exception. All other functions work independent of the state of the play option. Setting the play option won't start playing. Added multi-selection for the file requester.

18-Apr-94

if START AFTER LOAD was not set and the first file was started using CONT gadget or SPACE BAR, MIDI routes were not set correctly. This has now been fixed. Added font sensitivity, MidiPlay adapts itself to the screen font.

Cosmetic changes in the preferences window.

20-Apr-94

added f key for fast forward. Normal reset now overrides all filters. Directory buffer can be set off from menu. Playlist removing is now possible.

v2.11 24-Apr-94

load/save playlist added. ZOOM gadget added. MidiPlay can be quitted by running it again.

v2.12beta 30-May-94

illegal gadtools handling removed. Fixed erroneous time information which occurred when CIA timer was used. GUI modification. Added over 60 AREXX commands. Added PLAY option.

14-Jun-94

added AREXX commands: PATCHCHANNEL, VOLUMEDOWN, VOLUMEUP, SETVOLUMECHANGE. Added new tooltypes: PATCHCHANNEL, VOLUMECHANGE. Also FILENAMEREXX and FILETYPEREXX for user definable purposes like for displaying text files or unpacking non-midi files. SPEAK tooltype REMOVED! Fade now works as intended. Different MIDI files can now have different mapfiles. Solo channel bug fixed. Keyboard shortcuts for volume changing. Setting load .info on will set normal configuration values even if an icon is not found. MidiPlay.info tooltypes overriding/extending settings in the midiplay.config file are considered as configuration values.

26-Jun-94

Fixed FADE rounding error. Fixed a serious bug: some memory was freed twice.

v2.20beta 14-Jul-94

Added playlist, lyric and text event windows. Removed lyric entry in the SHOW menu. Removed the following ToolTypes: NOMON (replaced with MESSEAGEREXX script definition, this means: Upd not needed), PLSIZE (this is not needed anymore). Added ToolType LYRICTYPE. Added file information to the About requester. Added small delay before quitting (MidiOut couldn't send the reset message and crashed sometimes).

23-Jul-94

Fixed bug in About requester. Numeric pad now works.

v2.20 27-Jul-94

About requester works now!! If a track couldn't be loaded completely, counting lasted sometimes forever, not anymore.

31-Jul-94

Added tooltypes DIRBUFFER, FORCERESET. Fixed GUI. Fixed negative tempo case. Added ',' support for options/arexx commands which have used '|' before. VOLUMECHANGE requires now ':' instead of ','.

1-Aug-94

Bugs, bugs, bugs... Timer message not aborted at exit. Menus not freed. One requester not freed. And where do the rest hide?

v2.20a 17-Aug-94

Status cycle gadget was handled incorrectly. Skip value was read incorrectly. Now tries to read tooltypes from ENV:sys/def_midiplay.info before loading an icon of a MIDI file (sets default file values). Channel 0 has a special meaning for MUTE, NOMAPCHANNELS, VOLUMECHANGE and PATCHCHANNEL (it means that all channels will be affected). PATCHCHANNEL and VOLUMECHANGE ARexx commands can now have more than one setting. MERGE set right after loading.

21-Aug-94

SYSEX ARexx command. Menu available on all windows. Changed ARexx commands SETNOMIDISTART to SETMIDISTART and SETMIDIOUTPUT to SETNOMIDIOUTPUT (also the names in the menu bar have been changed). New or working ARexx commands: LOADMAP, LOADDRUMMAP, FILENAMEREXX, FILETYPEREXX, SETFORCERESET, SETLYRICTYPE, MESSAGEREXX, SETDIRBUFFER, SETSTATUSFIELD. Sysex messages printed.

22-Aug-94

Code for CLI options and tooltypes rewritten. Removed '-' from non-toggle options. Added '-' for MUTE and CLI option K (same as MAP).

24-Aug-94

Bugs...NOPC and NOPB ARexx commands didn't work properly. When playlist was removed, buffer value was set incorrectly. MidiPlay crashed sometimes when playlist was removed and forward was used. TEMPO 0 set now tempo to default value, i.e. 120 and TEMPO CHANGE is set off. Seconds left is now set correctly to zero, when count was not used. When Melody ARexx command is used, mute box is set now correctly. STOP TOGGLE works like the space bar.

v2.21a 6-Sep-94

Bug fix: if started from CLI without playlist, extra file was added. Int. reset moved before loading a new file. Fixed some incorrectly printed measure numbers.

18-Sep-94

Small changes in volumechange routines. Pattern gadget in file requester. Select requester calls now messengerexx script if it has been defined and illegal value is given.

v2.22 1-Oct-94

Added support for bank changes and external tone files in the mapfile.

4-Oct-94

Bug fix: volumechange via keyboard was broken.

8-Oct-94

Added sysex filter. Added CHANNELBYTE, BANKCHANGETYPE, NOSYSEX and CHANNELPC tooltypes. Added ARexx commands CHANNELBYTE, BANKCHANGETYPE, SETNOSYSEX and CHANNELPC.

v2.22a 22-Oct-94

Added mixer connection. CONNECTMIXER ARexx command.

v2.23 23-Oct-94

Changed keyboard handling, configurable keymapping now possible. For keyboard support added ARexx commands: MUTETOGGLE, STATUSFIELDUP, ACTIVATENEXTWIN, ACTIVATEDRUMGAD, ACTIVATEMELODYGAD, ACTIVATESOLOGAD, MELODYVOLUMEDOWN and MELODYVOLUMEUP.

v2.24 4-Nov-94

Added keyboard loading. New ARexx commands: LOADKEYBOARD, DEFAULTKEYBOARD. New tooltypes: KEYBOARD, DEFAULTKEYBOARD, PATTERN. Bank change type supports now both CC 0 and CC 32 messages.

v2.25 22-Nov-94

Window positions with open/closed and zoom information can be saved to an environment file. This file will be loaded automatically when started. Next number in the buffer can be seen in the main window's screen title.

23-Nov-94

Added NOTEON and NOTEOFF commands. Added PORTNAME tooltype. FORWARD command requires now value!! (fast forward speed is value*25%). FF gadget reacts immediately

when pressed down.

26-Nov-94

Added SETTINGS tooltip, configuration file should be named to midiplay.prefs and should be located in MidiPlay or in ENV:MidiPlay directory unless defined with SETTINGS.

v2.26 30-Nov-94

Added AppWindow to main and playlist windows. Icons dragged into these windows will be added to playlist. Added iconify command (+menu) and AppIcon. Icons dragged into AppIcon will be added to playlist. Double clicking on AppIcon opens earlier opened windows.

v2.27 14-Dec-94

Fixed a bug which changed the current file name before reading the correct icon file. It is possible to view only lyric events started by '<' mark by using LYRICTYPE=17 and lyric events started by '%' mark by using LYRICTYPE=33. This may however change in the future.

v2.28 20-Dec-94

Muting didn't work correctly (notes off sent on wrong channel).

v2.29 9-Jan-95

Bug fixes: icons without tooltips crashed MidiPlay. No midi output option works again. Meta events were printed over time information.

14-Jan-95

Bug fix: EOX messages were sent earlier with EOX byte.

v2.30 22-Jan-95

Tooltip MIDICLOCK added.

v2.30a 31-Jan-95

ARexx command symbols ON and TOGGLE work again. Serious quantization bug fixed. NOTE: tooltip is QUANTIZE (not QUANT as in previous documents)!

v2.31 5-Feb-95

Added more support for Mixer.

v2.31a 28-Feb-95

SYSEX messages were sent (again :-() incorrectly. Example3.rexx modified to work with powerpacked files.

>> v2.31b 26-Mar-95

Bug fix: Second MidiPlay crashed when one already running.

>> v2.31c 16-May-95

Addition: Titles of Karaoke MIDI files recognized.

>> v2.31d 6-Jun-95

Bug fix: Use of NOSYSEX option caused incorrect time on tracks which had sysex messages.

v2.40beta 5-Mar-95

Reqtools.library required no more, ASL file requester used instead (dirbuffer option is now obsolete). New select requester now with name selection. The edit requester in playlist window now like a file requester. Fixed bug which crashed MidiPlay when started and another MidiPlay was already running. New ARexx command SELECTNUMBER. SELECT command now has two possibilities: SELECT 0 (or SELECT, default) or SELECT 1 which activates the file name gadget.

16-Apr-95

Menus reorganized, many options now available also via menu. New feature: Intro play (songs played only 10 seconds each). New ARexx commands: SETPLAYSELECTED, SETINTROPLAY.

v2.50beta 17-May-95

All playing and loading routines now in a shared library called miditune.library.

19-May-95

ReqTools back as an option (tooltype REQTOOLS, ARexx command SETREQTOOLS). Intro time can now be set with INTROPLAYTIME.

20-May-95

New ARexx command SELECTBUFFER (selected number goes to the buffer). Rewritten some of the event handling code.

27-May-95

Fixed playlist handling code to work properly.

31-May-95

Added support for miditune.library's channel mapping with

CHANNELMAP command.

6-Jun-95

Fixed random number generating back to normal.

v2.51beta 21-Jun-95

New lyric window for better karaoke file support. New counting routine which checks also lyrics & titles.

7-Jul-95

Some lyric handling bugs fixed. Normal requesters support now also reqtools option. Sizes & places of windows saved now more accurately (font & border relative).

17-Jul-95

Resizable playlist and text event window. Text event window replaced with trackname window. More lyric bugs fixed. Also normal lyric text events collected if not karaoke file and count is set.

v2.52beta 20-Jul-95

PUBSCREEN tooltype to open MidiPlay on any existing public screen. Requesters opened on correct public screen. About requester shows also miditune.library version.

26-Jul-95

Fixed resizing bug when the width of the used font was smaller than 8. Lock option now correctly in menus. Top of the listview gadget in the playlist window now set according to the currently selected file.

31-Jul-95

Fixed tempo slider bug, when TEMPOCHANGE filter was set. Fixed forward gadget action back to normal, no more removal of all gadgets :). Fixed normal lyric handling when maximum number of rows exceeded. Made a hack for some karaoke files which have new line characters at the beginning of the lyric event.

v2.53beta 17-Aug-95

Fixed some lyric printing bugs when the window is quite small. Lyric starting point was not correctly set when START AFTER LOAD was used and playing was started with CONT command. Added LYRICFONT and LYRICCOLORS commands. WINDOW=ON back as default.

21-Aug-95

If the mixer bit was set in the environment file, the

mixer connection showed a wrong value (and the mixer wasn't opened at startup). Also if the select window bit was set, there were some serious troubles. These bugs are now fixed.

30-Aug-95

Fixed one enforcer hit in the default keyboard setting routine. Channelmaps cleared as default. Mixer should work now when connected.

v2.54beta 5-Sep-95

Changed string handling (edited strings back to original).

v2.55beta 11-Sep-95

QUIT ARexx command wasn't replied before exit. PATCHCHANNEL 0=0 command didn't work correctly. First parts of lyrics are now shown even if START AFTER LOAD is not set. In certain cases, only one paragraph was shown even if there was room for more.

v2.56beta 18-Sep-95

When the playlist is resized, the buttons are not vertically resized anymore. Tempo change filter works again.

v2.57beta 24-Sep-95

Added new commands: MERGETRANSPOSE and SENDMIDIDATA.

26-Sep-95

Name & copyright information were lost when select to buffer was made. Next file in the buffer was shown sometimes incorrectly. Lyric events from the previous file were sometimes left when the new file was started.

v2.58 30-Sep-95

Moving the tempo slider set tempo change filtering on. Final cleanup.

v2.58b 17-Oct-95

If the last file of the playlist was removed with the DEL button, the number of files in the playlist was incorrect. Fixed one lyric bug, which caused problems with some karaoke files.

v2.59 22-Oct-95

Fixed serious bugs in MUTECHANNELS and NOMAPCHANNELS string handling routines. Added new ARexx command STATUS.

v2.59b 23-Oct-95

Menu event without selected items caused Enforcer hits.

v2.59c 25-Oct-95

Fixed one illegal library call before exit.

v2.60 10-Nov-95

def_midisplay.info replaced with a user definable file (FILEDEFAULTS). Relative transposing with file icons is now optional (RELTRANSPOSE). If <file>.mid has corresponding configuration file <file>.cfg, it will be used instead of icon. A playlist can now have a configuration file (or icon tooltypes).

11-Nov-95

Commands available now also as tooltypes: TRANSPPOSEUP, TRANSPPOSEDOWN, TEMPOUP, TEMPODOWN, VOLUMEDOWN, VOLUMEUP.

v2.61 12-Nov-95

Added commands MIXERPATH, APPICONFILE, APPICONNAME, APPICONPOS.

v2.62 14-Nov-95

New buttons images for the main window. Reorganized the main window.

v2.63 19-Nov-95

Window for channel patching. FILEPATH command for the default MIDI file path.

v2.63b 20-Nov-95

Fixed bug: exit caused failure when MidiPlay was iconified.

1.161 AUTHOR

Programs, CDs, postcards, midi files, bug reports, requests, suggestions, questions to:

Janne Syväniemi
Lintuviidankatu 15 C 10
FIN-33340 TAMPERE
FINLAND

E-mail: csjasy@uta.fi
WWW Home Page: <http://www.uta.fi/~csjasy/>

Hardware: Amiga500+ with OS3.1, SupraTurbo28,
SupraDrive500XP 52Mb, 2Mb Fast Mem, MIDI interface, ROLAND D-10
and now also Roland SC-33 (THANKS TO CONTRIBUTORS!!), Microvitec

multiscan Monitor, Chinon HD Disk Drive

Software: Dr T's KCS v3.57

1.162 CREDITS

Thanks to:

Johnny B. Musik for SC-33,
Icon Guide
, MIDI files, help and all the
great ideas.

Erno Tuomainen for his BBS support, testing and ideas.

"Ozzy the Amiga Dood" Osborne for his ideas, map files and testing.

Claudio Bizzarri for testing (I can't dance :-))

Alain Valleton for his long enforcer hit reports (& The Visit)

Also to the authors of the following programs and libraries:

midi.library by Bill Barton

GadToolsBox by Jan van den Baard

reqtools.library by Nico François

Plus all the other people who have been interested.

REQUEST: If you make map files for different setups, please
share them with other people too by sending them
to me. I'll include them in future releases and
your name will be include in the list above.

1.163 TIPS FOR USERS

PROBLEM: I have two synths, one is a GM synth and another one is a guitar
module. How can I map correct patches to the guitar module on
channels 1 and 2 and let other channels to be played on the GM
synth without mapping?

SOLUTION: Make correct mapfiles for your guitar module and use the following
parameters in the configuration file:

MAP=ON

MAPFILE=GuitarModuleMap

NOMAPCHANNELS=3,4,5,6,7,8,9,10,11,12,13,14,15,16

PROBLEM: I have many MIDI files, but drums are always played too loud.

SOLUTION: If your synth plays drums on channel 10, put the following parameter in your configuration file:

```
VOLUMECHANGE=10:-20
```

This means that all volume messages are 20 lower on channel 10. Adjust the value (-20) so that drums are played as you like. You can also adjust the value manually using LSHIFT+0 (down) or RSHIFT+0 (up).

PROBLEM: How can I run MidiPlay from a directory utility like DirOpus, without window?

SOLUTION: Use command line "MidiPlay -w +S +X [<files>]". This plays files without window, starts immediately and exits after all files have been played. You can also use ARexx. Check out playonemidi.rexx example script.

PROBLEM: How can I easily set the MERGETRANSPOSE value to the value of normal transpose?

SOLUTION: You can use the STATUS ARexx command in an ARexx script. Request the value of transpose with STATUS TRANSPOSE and then use command 'MERGETRANSPOSE' result. Here is a complete example:

```
/* set mergetranspose = transpose */

options results
address 'MidiPlay_rexx'
'status transpose'
if rc == 0 then
  'mergetranspose' result
```

1.164 EXAMPLE 1

```
/* This is an example ARexx script for use with MidiPlay.
   Use the following parameter: MESSAGEREXX=ThisScript.rexx
   When MidiPlay shows a message on the screen, it calls ThisScript.
   Argument Text consists of the text of the message.
```

This particular example tries to speak the message. */

```
options results

parse arg Text

address command

'say' Text
```

1.165 EXAMPLE 2

```
/* This is an example ARexx script for use with MidiPlay.
   Use the following parameter: FILENAMEREXX=ThisScript.rexx
```

When MidiPlay has loaded a new file, it calls ThisScript.
Argument consists of drive, path, file and extension of the filename.

This particular example tries to view a text file with a name
'.../file.txt' when loaded file was '.../file.mid'.

*/

options results

```
parse arg Drive ',' Path ',' File ',' Extension .
```

```
if Path == "" then Path2 = Path
else Path2 = Path || '/'
```

```
Ext2 = "txt"
```

```
Whole = '' || Drive || Path2 || File || '.' || Ext2 || ''
```

```
if exists(Whole) then
  'run >nil: more' '''Whole'''
```

1.166 EXAMPLE 3

/* This is an example ARExx script for use with MidiPlay.
Use the following parameter: FILETYPEREXX=ThisScript.rexx
When MidiPlay cannot load a file, because it is not a MIDI file, it
calls ThisScript. Argument consists of drive, path, file and extension
of the filename.

This particular example tries to unarchive a file with extension lha
to a directory ram:t:rxmidi. If unarchiving is succesful, the first .MID
file is played by sending command 'FILE first.MID' to MidiPlay.

*/

```
lhadir = 'C:'
```

options results

```
parse arg Drive ',' Path ',' File ',' Extension .
```

address command

```
if ~show('L',"rexxsupport.library") then do
  if ~addlib('rexxsupport.library',0,-30,0) then
    exit 10
end
```

```
if ~exists('t:rxmidi') then
  'mkdir t:rxmidi'
```

```
if Path == "" then Path2 = Path
else Path2 = Path || '/'
```

```
Whole = ''' || Drive || Path2 || File || '.' || Extension || '''
```

```
"delete >nil: t:rxmidi/#?"

if Extension='lha' | Extension='LHA' then
  lhadir'lha >nil: -x0 -q e 'Whole' t:rxmidi/'
else do
  say 'Not a MIDI file' Whole
  RETURN
end

dir1=showdir('t:rxmidi/', 'f', '@')
dir1=insert(dir1, '@')
dir2=upper(dir1)

position=index(dir2, '.MID')
position3=lastpos('@', dir2, position)
position2=index(dir2, '.MID', position)
filenam=substr(dir2, position3+1, (position2-position3)+3)

address "MidiPlay_rexx"
'FILE t:rxmidi/'filenam
```

1.167 MIDIPLAY TOOLKIT

MidiPlay ToolKit is a program collection to extend the capabilities of MidiPlay. Currently there are a simple Mixer, KeyboardEditor, Midi2Text and Text2Midi convertors in the toolkit. Later there will be a singer, a sample player, and a map file editor and more...

You can get MidiPlay ToolKit by becoming a registered user of MidiPlay. You become a registered user of MidiPlay by sending me a gift and necessary information about yourself and your setup. Recommended gift is a long play CD (only GOOD music!).

Mixer v1.2: Sliders for changing the volume and pan/reverb/chorus values on channels. Meters to show active channels. Instrument numbers also shown.

KeyboardEditor v1.4: Every rawkey message with any qualifier combination can be mapped to different command or to an ARexx scriptfile. Different messages can be made for even keydown and keyup messages or repeat keys. MidiPlay will support note on and note off messages in future. Then you can make your keyboard file so that different keys play different pitches and maybe with some qualifier combination different chords.

Text2Midi v1.0 and Midi2Text v1.0: Midi2Text utility converts MIDI files to ASCII text format. The text files can then be edited with a normal text editor and converted with the changes back to a MIDI file with Text2Midi

utility.

1.168 How to adjust MIDI files using icons

How do I adjust the Replay of my Midi files using the Midi file Icons ??????

On a Amiga you can for all files create an Icon which is displayed on the Workbench as a picture. The easiest way to create Icons is to use a prg like Dir Opus, in that prg you can simply mark your Midifiles and click the button ADD ICON and Dir Opus then creates an icon for the Midifile.

You can then edit your Icon from Dir Opus just by clicking twice on the file that has the same name as your Midifile + the ext .info.

The Icon picture then pops up, and then you can click new and in the bottom window you can input your command and finish with Enter, then the command is moved from the bottom window to the top window and is used by Midiplay when the Midifile is loaded.

If you dont have Dir Opus you can open the Icon from WB and choose information and edit from there.

You can always reedit your commands by opening the Icon and click on the command you want to edit, by clicking the commands is displayed in the bottom window and you can alter the command, when finished always remember to use Enter, and done with editing you must remember to save the Icon.

What can I actually adjust in those Icons ??

You can really adjust ALL commands that Midiplay understands but the most important are these !!

»~Melody~Channel~::~

»~Transpose~::~

»~Tempo~Change~::~

»~Program~Change~::~

»~Patch~Channel~::~

»~Volume~Change~::~

I have in this documentation written all Commands with Capital letters, but Midiplay don't care so you are allowed

to use both or mix like you prefer.

1.169 Melody channel

Command MELODY=Value

This Commands tells Midiplay on which channel your Melody are playing and the value are simply the MIDI Channel number

You simply write in you Command line this:

MELODY=Value and the value can be positive/negative 1-16

If the value are Positive the Melody will be muted, and will not play when the Replay is started.

Ex:

```
MELODY=4
```

The Melody voice is on channel 4 and not played.

When you give this command to Midiplay then this info is stored to the F8 key and you can now anytime, get the melody to play just pressing the F8 key.

Remember to put SOFT=ON in Midiplay config if you want Midiplay to send out all Mididata on the muted melody channel, all Mididata except note on/off is then sent, and the melody will sound with correct voice when you unmute.

SOFT=ON can also be a command line in the MIDI file Icon if you prefer that.

If the value is Minus=Negativ then the Melody is played when you start the Replay.

Ex:

```
MELODY=-4
```

The Melody is on channel 4 and is played.

When you give this command to Midiplay then this info is stored to the F8 key and you can now anytime, mute the melody just pressing the F8 key.

In both cases repeated pressure on the F8 key toggles between muted and unmuted Melody voice.

So you must understand that the Positive or negative value only is there to tell Midiplay how to play when first starting the Melody.

1.170 Transposing

Command `TRANPOSE=value`

The value can be 1-12 both Positive and Negative.

The command is used for transposing the Melody.

If the Drumchannel is defined then the drumchannel is not Transposed.

Ex: `TRANPOSE=2`

This tells Midiplay to transpose the Midifile 2 half notes up.

Ex: `TRANPOSE=-4`

This tell Midiplay to transpose the Midifile 3 half notes down.

If you have given `transpose=2` in the config file the the command `TRANPOSEUP=2` gives the result $2+2 = 4$ half notes up and the `TRANPOSEUP=-4` gives the result $2-4 = 2$ half notes down.

This means that the tranpose command given in the Midifile Icon are always active no matter what transpose you have given elsewhere, and thats good because if you catch a cold and need to transpose all Midifile half a note to sing with you cold voice you can simply `transpose=-1` in the Config file and all your Midifiles are transposed half a note down.

1.171 Tempo changing

Command `TEMPO=beats pr measure`

Values can be between 40-240 or 0

Ex: `TEMPO=136`

This command tell's Midiplay to play the Midifile in tempo 136. BPM

If you use this command be sure there is no tempo change in your Midifile because using the command Midiplay ignores all tempo settings in the MIDI file.

Setting `TEMPO=0` means use default tempo = 120

1.172 Program change on a channel

```
Command CHANNELPC=Channel:Value  
Command CHANNELPC=Channel:Bbank:Value
```

Channel can be between 1-16
Bbank can be between 0-127
Value can be between 0-127

Ex:

```
CHANNELPC=4:24
```

You tell Midiplay to send a Program Change 24 on channel 4

Program change 24 will on General Midi choose a Nylon Guitar, but Nylon Guitar on the Display say 25 which is a mistake and don't match reality. In Midi everything starts with 0 which means that 1 Grand Piano really is prg change 0 etc.

You must therefor always subtract 1 if you want a certain prg change on you synth if you look it up in the manual.

You can make as many lines with channelpc as you wish, but its also possible to make several in the same command line like:

Ex:

```
CHANNELPC=4:24,5:34,9:10
```

Just seperate the channelpc's using a ,

Using channelpc the program change on that channel are ignored and only the given will be used.

If your Midifiles contains a Reset SYSEX you have to filter sys x because the prg change in channelpc are send and the prg change on the channel are ignored but the reset sets all parts to G Piano so you wont get the Prg change you intended.

You can use the command NOSYSEX=ON in the Icon then if you normally want it on you can input the command in the file defaults file NOSYSEX=OFF this way every time a new file is loaded the filter off.

You also now can do a bank change using Channelpc first you have to define in your config file how Midiplay should send Bank change.

You must use the command bakchangetype

Ex:

```
BANKCHANGETYPE=0  sends Controlchange 0 + value followed
                    by control change 32 0
BANKCAHNGETYPE=32 sends Controlchange 32 + value followed
                    by control change 0 0
```

Please look into your manual to se which way your synth should be told to change banks.

If you want to change to a 12 string guitar on a GS module you must send bank change 8 and prg change 25 and you do like this:

```
CHANNELPC=4:B8:25
```

You must always give the Bank change as B+value.

Also by using Channelpc you can send SYS EX files to your synth, you must use a sysx dump prg to recieve the sysx dumps and then save them under a filename then using Channelpc you can send these dumps back

lets say you have a dump called mydump you write like this

```
CHANNELPC=1:Tmydump
```

Remember always to send Sysex dumps on channel 1

1.173 Channel patching

```
Command  PATCHCHANNEL=Channel=Channel
```

Channel can have value between 1-16

You tell Midipay that all on first given channel must be send on second given channel

Ex:

```
PATCHCHANNEL=1=10
```

All Events on channel 1 is now send on channel 10
I use this one a lot because many Midi Creators uses 4 drum beats on channel 1 to count in, and my synth need them on channel 10 to play drums.

I have standardised all my Midfiles to always play the Melody on channel 4 and the Bass on channel 2.
I play the Midifile and notes on which channels melody and bass play, and if its not channel 4 and 2 I patch the file.

Wee assume we have a Midifile where Melody plays on channel 9 and the BASS plays on channel 6 and wee then write this in the Midifile Icon.

```
PATCHCHANNEL=9=4,4=9,6=2,2=6
```

Notice that every patch is seperated using a ,
In this patch we let channel 9 play on channel 4 therefor I say let then channel 4 play on channel 9 and the same for the BASS. I always do this channel switch this way I'm sure nothing goes wrong and data is not mixed.

Now Melody=-4 will work on the melody voice because we have Patched = moved it to channel 4

All other commands also works on the given channel even if you have patched around.

1.174 Volume Changing

```
Command VOLUMECHANGE=Channel:Value
```

Channel can be between 1-16

Value can be between Plus/Minus 0-127

Ex:

```
VOLUMECHANGE=4:25
```

Here we tell Midiplay to raise volume on channel 4 with 25

```
VOLUMECHANGE=4:-25
```

Here we tell Midiplay to lower volume on channel 4 with 25

The Value we give is always added to the volume information in the Midifile. If a volume info is not found on the channel the value given is added to the default value 100

This means giving a volumechange it is added to all volumes on the given channel.

ADDING a negative value lowers the volume.

Also here you can give several channels.

```
VOLUMECHANGE=4:20,3:-10,6:5,10:-5
```

Again you seperate using the , and here we say:

```
ADD 20 on channel 4  
ADD -10 on channel 3  
ADD 5 on channel 6
```

ADD -5 on channel 10

Another fine feature to use is this if your Midifile plays to loud or to weak on all channels then use the command like this:

VOLUMECHANGE=0:25 this raises volume on all channels by 25
VOLUMECHANGE=0:-25 this lowers volume on all channels by 25

Then afterwards you could tell Midiplay to ex raise volume on channel 4 by 25.

I use that often if I have a Midifile where all volumes are on 127 and the melody is to low to be heard, then I lowers all channel and raises the melody channel.

Ex:

VOLUMECHANGE=0:-30 lowers all channels by 30
VOLUMECHANGE=4:30 raises volume on channel 4 by 30.
